A LV-MPC STRATEGY FOR TRAJECTORY TRACKING IN BATCH PROCESSES

José L. Godoy[(a)], Jacinto L. Marchetti[(b)], Jorge R. Vega[(c)]

[(a),(b),(c)] Instituto de Desarrollo Tecnológico para la Industria Química (INTEC, CONICET – UNL),
Güemes 3450 (3000), Santa Fe, Argentina
[(c)] Departamento de Ingeniería Eléctrica, Facultad Regional Santa Fe (FRSF-UTN),
Lavaisse 610 (3000), Santa Fe, Argentina

[(a)] jlgodoy@santafe-conicet.gov.ar, [(b)] jlmarch@santafe-conicet.gov.ar, [(c)] jvega@santafe-conicet.gov.ar

ABSTRACT

A constrained latent variable model predictive control (LV-MPC) technique is proposed for trajectory tracking in batch processes. The controller allows the incorporation of constraints on the process variables and is designed on the basis of multi-way principal component analysis (MPCA) of a batch data array that is rearranged by means of a regularized batch-wise unfolding approach. The LV-MPC formulation includes a novel prediction stage and is offset-free. The controller parameters are calculated on the basis of the identified latent model. The main advantages of LV-MPC over other MPC techniques are: (i) a relatively small dataset is required (e.g., around 10-20 batch runs), (ii) nonlinear processes can efficiently be handled algebraically through MPCA models, and (iii) the tuning procedure is simple. The proposed constrained LV-MPC technique is numerically tested through a benchmark process that has been used in previous LV-MPC formulations.

Keywords: multi-way principal component analysis, latent variable model, constrained predictive control.

1. INTRODUCTION

In general, it is accepted that phenomenological inner behaviors of many batch processes (e.g., pharmaceutical or biotechnological processes) are not well understood at present, and for this reason, fundamental or knowledge driven models are difficult to be obtained. An additional difficulty that complicates the development of such models is the small production scales associated to pharmaceutical products, especially when compared to the large amount of product obtained in many chemical and petrochemical processes. For these traditional processes, a large amount of knowledge-driven models has been developed and extensively used over the last decades. However, for any process, an available data-driven model can help to better understand the process and therefore to design operation policies that contribute to improve the process performance and the quality of the final products. In contrast, Troup and Georgakis (2013) suggested that a data-driven model can be more useful than a knowledge-driven model because the former is often more suitable for a quick deployment in process optimization and on-line control tasks, particularly for the case of pharmaceutical processes.

Batch and semi-batch processes are used in many industries because of their flexibility to manage several grades and types of products. In these processes, one of the requirements to achieve appropriate final quality specifications and adequate operation is to track reference signals that have been determined by an independent optimization stage. Proportional-integral-derivative (PID) controllers are by far the most common approach used in industry. However, batch processes usually exhibit large time constants and time varying dynamics, and sometimes it is necessary to track complex set-point trajectories. Under these scenarios, standard PID controllers might render a poor performance. Conventional PID controllers have been modified to mitigate some of their classical deficiencies. For instance, some improved PID controllers are: PID-feed forward controllers (Clarke-Pringle and MacGregor 1997), adaptive PID controllers (Lontra 1991), and self-tuning PID controllers (Altintena, Ketevanlioğlua, Erdoğana, Hapoğlub and Alpbaz 2008). In addition, advanced control approaches based on nonlinear theoretical models of the batch process have been proposed to improve PID performance, as for example globally linearizing control (Kravaris, Wright and Carrier 1989; Kravaris and Saroush 1990) and generic model control (Cott and Macchietto 1989; Wang, Pla and Corriou 1995; Aziz, Hussain and Mujtaba 2000).

Another advanced control strategy used for complex multivariate processes is Model Predictive Control (MPC). Garcia (1984) implemented a MPC strategy for the temperature control of synthetic rubber production in a semi-batch process; Gattu and Zafiriou (1992) extended the work of Garcia (1984) by incorporating Kalman filter estimation. Multi-input multi-output (MIMO) MPC was addressed by Peterson, Hernandez, Arkun and Schork (1992) for the control of temperature and average molecular weight in the solution polymerization of methyl methacrylate (MMA). The major advantage of MPC is its capability to fulfill general control objectives (including economic aspects), by simultaneously taking into account a dynamic simplified model of the plant, constraints, and stability requirements (Camacho and Bordons 2004; Rawlings and Mayne 2009). This is probably the main reason why MPC is one of the advanced control strategies often used in large-scale industries.

First industrial applications of MPC have solved the dynamic and economic problems through a two-layer optimization approach. However, in the last years, there is a tendency to include the economic objectives in the MPC controller. Several economic formulations of MPC that get over the standard set-point-tracking formulation have been presented (Ferramosca, González, Limon, Bustos, Godoy and Marchetti 2014). These controllers integrate the economic cost function of the Real Time Optimization (RTO) as an additional stationary cost to the dynamic quadratic cost of a MPC for set-point tracking (Zanin, Tvrzska de Gouva and Odloak 2002). For this reason, the process control variables attempt to minimize the (possibly non-linear) economic cost.

A main goal of most MPC software suppliers is to provide simple tuning algorithms and modest experimental design for process identification that facilitate the dissemination of their products. In this regard, the latent variable model predictive control (LV-MPC) technique is emerging as a viable alternative to be implemented in industry. Troup and Georgakis (2013) highlighted the interest in LV-MPC, in particular for batch processes, because it is an alternative to Nonlinear Model Predictive Control (NMPC), with the advantage of not using non-linear functions. LV-MPC algorithms are based on Principal Component Analysis (PCA) models developed on batch data arrays (Nomikos and MacGregor 1994), where the prediction of the future trajectories is accomplished by using statistical latent variable missing data imputation methods (Nelson, Taylor and MacGregor 1996). Flores-Cerrillo and MacGregor (2005) have developed a version of LV-MPC for batch processes using a PCA model and Yu and Flores-Cerrillo (2013) have proposed a design methodology to select the corresponding parameters. Another LV-MPC technique based on multiphase modeling of a Batch-Wise Unfolding (BWU) of data arrays has been proposed in MacGregor, Bruwer and Golshan (2009) and in Golshan, MacGregor, Bruwer and Mhaskar (2010), which involves an additional modeling step and furthermore it includes future set-points in the predictive model. The irregularity of the set-point trajectory improves the conditioning of the predictive model. Therefore, this technique is highly dependent on the used trajectory. The BWU modeling approach addresses the nonlinearity and time varying properties of the batch process. However, it needs a large number of batch runs in the training period. This has motivated the so-called Regularized Batch-Wise Unfolding (RBWU) to decrease the number of collected batches necessary to identify a model (Golshan, MacGregor and Mhaskar 2011). This modeling alternative has been incorporated in the LV-MPC methodology proposed in this article.

Another important point to be highlighted is related to the manipulated variables. Golshan, MacGregor, Bruwer and Mhaskar (2010) have shown that identical performances can be achieved through control strategies implemented in the latent variable space and in the manipulated variable space. However, the control in the manipulated variable space is preferable because the optimization variables can be directly constrained. For this reason, the formulation proposed in this paper presents explicit constraints in the space of the manipulated variables.

Therefore, the main difference in the formulation presented in this work, compared to previous techniques, is that the predictive model does not include the set-point trajectory, making it more flexible. Also, another important contribution of the work is the proposal of a new LV-MPC formulation for tracking, which is offset-free and subject to constraints, where the prediction is done in a rather different manner from other authors (MacGregor, Bruwer and Golshan 2009; Golshan, MacGregor, Bruwer and Mhaskar 2010; and Golshan, MacGregor and Mhaskar 2011).

The rest of the work is organized as follows. In the next section, the constrained LV-MPC for batch processes is presented. Section 3 is devoted to analyze the study cases. The method is tested on a simulated batch reactor, which serves as a benchmark because it has previously been used to evaluate LV-MPC techniques. The paper ends with some conclusions.

2. CONSTRAINED LV-MPC FOR BATCH PROCESSES

Consider first a PCA model of a batch process that has been developed on the basis of a batch dataset. Assume that a set-point trajectory has been defined for the process. Then, the main objective is to track the specified trajectory in a new batch run, which can additionally be affected by disturbances. In what follows, the model calibration and the controller design are described.

2.1. Rearrangement of the batch datasets

Consider the i-*th* batch run. For the sample time *k*, the measurement vector is defined as follows:

$$\mathbf{x}'_{i,k} = \left[\mathbf{y}'_{i,k}\ \mathbf{u}'_{i,k}\ \mathbf{d}'_{i,k}\right]_{(1\times n)}, \tag{1}$$

where $\mathbf{y}_{i,k}(ny\times 1)$, $\mathbf{u}_{i,k}(nu\times 1)$ and $\mathbf{d}_{i,k}(nd\times 1)$ are the controlled, manipulated, and measured variables, respectively; and $n = ny + nu + nd$. Particularly, $\mathbf{d}_{i,k}$ is a vector of on-line measurements (e.g., pressure, temperature, stirring rate, flow rates, etc.) that can be incorporated to give information on disturbances and process changes. The data collected from a batch process are arranged in a 3-dimensional array (or cube) where for I batch runs ($i = 1,\ldots,I$), the trajectories of n variables are measured over K time intervals ($k = 1,\ldots,K$). Latent variable modeling of these data involves unfolding the data array into a 2-dimensional matrix and then modeling the variation in this matrix. The main difference among the existing approaches (Golshan, MacGregor and Mhaskar 2011) stems from the strategies utilized to construct a 2-dimensional array (a matrix) from the 3-dimensional data array. Nomikos and MacGregor (1994) suggested that the BWU approach is the most logical way for modeling the differences among batches. In the BWU approach, all the variables at different sample times are put beside each other and each batch history constitutes one observation or row in the unfolded matrix, i.e. :

$$\mathbf{X}_B = \left[\mathbf{X}_1 \cdots \mathbf{X}_k \cdots \mathbf{X}_K\right]_{(I\times Kn)}, \text{ with } \mathbf{X}_k = \begin{bmatrix} \mathbf{x}'_{1,k} \\ \vdots \\ \mathbf{x}'_{I,k} \end{bmatrix}_{(I\times n)}, \quad (2)$$

In $\mathbf{X}_B$, data on each variable at all time intervals are included in a row. Thus, a PCA model of $\mathbf{X}_B$ is capable of explaining the time varying and nonlinear characteristics of the batch. The calibration data set can be obtained from the previous batches run in normal conditions, augmented with additional batches that have designed according to identification experiments, in order to provide more information on the causal relationships at every time interval.

In this work, RBWU is used to produce a regularized version of the PCA model of $\mathbf{X}_B$ from a reduced number of batch runs (Golshan, MacGregor and Mhaskar 2011). This approach unfolds batch-wise but also repeats each batch row L times, each time shifted by one additional sampling interval, as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_{K-L} \\ \mathbf{X}_2 & \mathbf{X}_3 & \cdots & \mathbf{X}_{K-L+1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{X}_{L+1} & \mathbf{X}_{L+2} & \cdots & \mathbf{X}_K \end{bmatrix}_{(I(L+1)\times(K-L)n)}. \quad (3)$$

The parameter L is the number of shifts in matrix **X**. If L = 0 (no shift), the unfolding is simply BWU (Eq. 2). But if a small number of shifts is used (e.g., L/K < 0.05), this approach will retain most of the advantages of the BWU approach (by capturing the time varying non-linear behavior). However, at each time interval, the model will be averaged over L consecutive time periods thereby restoring some of the advantages of the Observation-Wise with Time-lag Unfolding (OWTU). This modeling strategy aims at capturing the major benefits of the two previously mentioned modeling approaches, while avoiding the problems related to each one. A PCA model based on the unfolded matrix **X** (Eq. 3) that contains mean-centered and scaled data produces a reduced dimension latent model, as follows:

$$\hat{\mathbf{X}} = \mathbf{XPP}', \quad (4)$$

where $\mathbf{P}$ $((K-L)n\times A)$ is the loading matrix, with A being the number of retained latent variables (Godoy, Vega and Marchetti 2014). To produce an adequate model, the value of A must be smaller than or equal to the number of batches I that are available in the calibration dataset (i.e., $A \le I$). The mean centering procedure automatically removes the average trajectories of all variables and hence also removes the main nonlinearities related to the absolute values of such variables. Then, the application of PCA on these deviated data provides different loadings at each time, thereby modeling the instantaneous covariance structure and its changes along the time. Therefore, it provides a locally linearized model of the covariance structure of the variables at every point. As a consequence, this PCA model captures the time-varying properties throughout the batch as a locally linear model at each point along the batch time.

The covariance of the time-shifted matrix **X** is the average of the covariance matrix of batch-wise unfolded matrix $\mathbf{X}_B$ over (L+1) consecutive sample times (Golshan, MacGregor and Mhaskar 2011). Thus, the resulting covariance matrix is a regularized version of the original covariance matrix. Note that the loading matrix **P** of the PCA model built for the **X** matrix is a regularized version of the loading matrix for the $\mathbf{X}_B$ matrix, where L is the moving average window length. The value of L should be chosen as the maximum value such that the process variability is maintained and L/K <0.05, while the variability by common causes is reduced (signal smoothing).

2.2. Identification

The calibration data can be taken from batches that have been run under normal conditions, and augmented with additional batches carried out according to experiments properly designed for identification purposes. These experiments must provide information about the causal relationships between the manipulated and controlled variables, at every time interval. The direct identification approach based on closed-loop data is used in this study. Closed-loop identification is preferred over open-loop identification for batch processes in order to maintain the process close to its desired trajectories and to minimize the variations of the final product quality.

A dither signal in the form of a Random Binary Sequence (RBS) is added to the manipulated variable trajectories coming from an existing proportional-integral (PI) controller to provide some additional excitation to the process. The RBS signal is chosen to have its switching frequency in a suitable range (~1/3 of the dominant time constant of the process). The closed-loop design of the identification experiments for identifying models relating time-varying batch trajectories have previously been used in Golshan, MacGregor, Bruwer and Mhaskar (2010). The designed RBS signals simply improve the causal relationships between the manipulated and controlled variables along their trajectories. The historical batch data are also important in providing models for the effects of inherent disturbances in the batch process and their influence on the behavior of the evolving trajectories.

2.3. Prediction of future trajectories

Applying PCA to the unfolded matrix allows modeling the time-varying and nonlinear behavior of the batches as a local linear model at every sample time. We take this local characteristic of the model to propose the predictor described below.

Suppose a multi-way PCA model is developed based on a regularized batch-wise unfolded dataset (Eq. 4). Therefore, each row $\mathbf{x}'$ of the unfolded **X** matrix (Eq. 3) corresponds to the data from one complete batch, which can be modeled as: $\hat{\mathbf{x}}' = \mathbf{x}'\mathbf{P}\mathbf{P}'$. Assume that a new batch is currently at sample time *k*. Then, the variables in that batch, $\mathbf{x}'$, can be partitioned into four terms (distant past, recent past, near future, and distant future) as follows:

$$\mathbf{x}' = \Big[\underbrace{\mathbf{x}'_1 \cdots \mathbf{x}'_{k-2PH-1}}_{\mathbf{x}'_{p2,k}} \underbrace{\mathbf{x}'_{k-2PH} \cdots \mathbf{x}'_{k-1}}_{\mathbf{x}'_{p1,k}}$$
$$\underbrace{\mathbf{x}'_{k} \cdots \mathbf{x}'_{k+PH-1}}_{\mathbf{x}'_{f1,k}} \underbrace{\mathbf{x}'_{k+PH} \cdots \mathbf{x}'_{K-L}}_{\mathbf{x}'_{f2,k}}\Big],$$
$$\mathbf{x}' = \left[\mathbf{x}'_{p2,k}, \mathbf{x}'_{p1,k}, \mathbf{x}'_{f1,k}, \mathbf{x}'_{f2,k}\right], \tag{5}$$

where *PH* is the selected prediction horizon. The corresponding loading matrix **P** (see Eq. 4) can also be separated in the same way as the $\mathbf{x}'$ vector as follows:

$$\mathbf{P}' = \left[\mathbf{P}'_{p2,k} \quad \mathbf{P}'_{p1,k} \quad \mathbf{P}'_{f1,k} \quad \mathbf{P}'_{f2,k}\right], \tag{6}$$

where coefficients in $\mathbf{P}'_{p1,k}$ and $\mathbf{P}'_{f1,k}$ account for the correlations between measurements, from *k*-2*PH* to *k*+*PH*-1. Then, the near future process variables $\mathbf{x}_{f1,k}$ can be estimated using missing data imputations, as follows:

$$\hat{\mathbf{x}}_{f1,k} = \mathbf{P}_{f1,k}\left(\mathbf{P}'_{p1,k}\mathbf{P}_{p1,k}\right)^{-1}\mathbf{P}'_{p1,k}\,\mathbf{x}_{p1,k}\,, \tag{7}$$

where $rank(\mathbf{P}'_{p1,k}\mathbf{P}_{p1,k}) = rank(\mathbf{P}'\mathbf{P}) = A$. This prediction is based on the following latent relationship: $\left[\mathbf{x}'_{f1,k}, \mathbf{x}'_{p1,k}\right] = \mathbf{t}'_k\left[\mathbf{P}'_{f1,k}, \mathbf{P}'_{p1,k}\right]$, where the latent variables, $\mathbf{t}_k$, could be estimated by using the known part of the data, i.e. $\hat{\mathbf{t}}_k = \left(\mathbf{P}'_{p1,k}\mathbf{P}_{p1,k}\right)^{-1}\mathbf{P}'_{p1,k}\,\mathbf{x}_{p1,k}$; and hence, $\hat{\mathbf{x}}_{f1,k} = \mathbf{P}_{f1,k}\hat{\mathbf{t}}_k$ (Godoy, Vega and Marchetti 2014). Therefore, we obtain the prediction of near future behavior $\mathbf{x}_{f1,k}$ by using the recent past data $\mathbf{x}_{p1,k}$. The inclusion of $\mathbf{d}_k$ in the modeled measurements (Eq. 1) improves the consistency of the correlation model (Eq. 4) and therefore the predictive model (Eq. 7).

The idea behind this partition is to create a (short) moving window for prediction, in order to avoid local linear model unreliability. This data influence window is determined (only) by a short portion of the complete batch vector **x**. That is, only the recent past will be used to predict the near future behavior, by means of a local dynamic model. Furthermore, given that a local dynamic model is used, both, the distant past and future behavior are not taken into account for prediction. The use of this partition (not the partition itself) constitutes a novelty in contrast to the formulation presented in Golshan, MacGregor, Bruwer and Mhaskar (2010), which considers the set-point trajectory, the recent past and the distant past for the near future prediction. This later formulation derives in a low sensitive prediction, given that too much past information is used.

2.4. Constrained LV-MPC for trajectory tracking

The idea now is to propose a MPC strategy for trajectory tracking, taking advantage of the LV model presented before. The near future outputs, at time *k*, $\mathbf{y}'_f = \left[\mathbf{y}'_k \cdots \mathbf{y}'_{k+PH-1}\right]$, will be predicted by means of the recent past data, $\mathbf{x}_{p1,k}$, and future control inputs, $\mathbf{u}'_f = \left[\mathbf{u}'_k \cdots \mathbf{u}'_{k+PH-1}\right]$, using missing data imputation (where $\mathbf{y}_f$ and $\mathbf{u}_f$ are included in $\mathbf{x}_{f1,k}$). To do that, the combined vector $\mathbf{x}'_{p,k} = \left[\mathbf{x}'_{p1,k}, \mathbf{u}'_f\right]$ is first defined, and then, the appropriate model partition corresponding to the this vector, $\mathbf{P}'_{p,k} = \left[\mathbf{P}'_{p1,k}\ \mathbf{P}'_{u,k}\right]$, is computed. Matrix $\mathbf{P}_{u,k}$ (*nu PH* × *A*), in the later partition, contains the rows of $\mathbf{P}_{f1,k}$ ($PH\,n \times A$) corresponding to $\mathbf{u}_f$. This way, the near future outputs can be predicted by:

$$\hat{\mathbf{y}}_f = \mathbf{P}_{y,k}\left(\mathbf{P}'_{p,k}\mathbf{P}_{p,k}\right)^{-1}\mathbf{P}'_{p,k}\,\mathbf{x}_{p,k} = \mathbf{C}_k\begin{bmatrix}\mathbf{x}_{p1,k}\\ \mathbf{u}_f\end{bmatrix}, \tag{8}$$

where $\mathbf{P}_{y,k}$ is composed by the rows of $\mathbf{P}_{f1,k}$ corresponding to $\mathbf{y}_f$. Although this prediction model captures the main relationship between future outputs and recent past data together with future control inputs, both are normalized (i.e. centered by their means and scaled by their deviations). Thus, in order to obtain the future output prediction vector denormalized, it should be scaled back. This way, an output prediction in the original units is given by:

$$\hat{\mathbf{y}}_f = D_{\mathbf{y}_f}\left[\mathbf{C}_{1,k}\,\mathbf{C}_{2,k}\right]\begin{bmatrix}D^{-1}_{\mathbf{x}_{p1,k}}\left(\mathbf{x}_{p1,k} - \bar{\mathbf{x}}_{p1,k}\right)\\ D^{-1}_{\mathbf{u}_f}\left(\mathbf{u}_f - \bar{\mathbf{u}}_f\right)\end{bmatrix} + \bar{\mathbf{y}}_f\,, \tag{9}$$

where $\mathbf{C}_k = \left[\mathbf{C}_{1,k}\,\mathbf{C}_{2,k}\right]$, and $D_{\mathbf{x}_{p1,k}}$, $D_{\mathbf{u}_f}$, $D_{\mathbf{y}_f}$ are the deviations diagonal matrices and $\bar{\mathbf{x}}_{p1,k}$, $\bar{\mathbf{u}}_f$, $\bar{\mathbf{y}}_f$ are the means vectors, both corresponding to the time interval $k \cdots k + PH - 1$. Finally, as usual in MPC closed-loop, a correction term is added to the prediction, to account for the feedback:

$$\begin{aligned}\hat{\mathbf{y}}_f = D_{\mathbf{y}_f}\left[\mathbf{C}_{1,k}\,\mathbf{C}_{2,k}\right]\begin{bmatrix} D^{-1}_{\mathbf{x}_{p1,k}}\left(\mathbf{x}_{p1,k} - \bar{\mathbf{x}}_{p1,k}\right) \\ D^{-1}_{\mathbf{u}_f}\left(\mathbf{u}_f - \bar{\mathbf{u}}_f\right)\end{bmatrix} + \bar{\mathbf{y}}_f \\ + \mathbf{K}(\mathbf{y}_k - \hat{\mathbf{y}}_k)\end{aligned} \qquad (10)$$

where $\mathbf{K} = [\mathbf{I}_{ny} \ldots \mathbf{I}_{ny}]'$ $(PH\,ny \times ny)$, $\mathbf{y}_k$ is the current output, and $\hat{\mathbf{y}}_k$ is the current prediction in original units; which can be estimated as follows:

$$\hat{\mathbf{y}}_k = D_{\mathbf{y}}\mathbf{P}_{1,k}(\mathbf{P}'_{p1,k}\mathbf{P}_{p1,k})^{-1}\mathbf{P}'_{p1,k}D^{-1}_{\mathbf{x}_{p1,k}}(\mathbf{x}_{p1,k} - \bar{\mathbf{x}}_{p1,k}) + \bar{\mathbf{y}}_k \text{ ,(11)}$$

where $\mathbf{P}_{1,k}$ is composed from rows of $\mathbf{P}_{f1,k}$ corresponding to $\mathbf{y}_k$; and $D_{\mathbf{y}}$ and $\bar{\mathbf{y}}_k$ are the deviations diagonal matrix and the means vector, respectively.

Now, the future input vector, $\mathbf{u}_f$, can be expressed in terms of the control moves, $\Delta\mathbf{u}_f$, which constitutes an appropriate practice in MPC to obtain an offset-free formulation (González, Adam and Marchetti 2008). Assuming a control horizon *CH*, $\mathbf{u}_f$ can be written as follows:

$$\begin{aligned}\mathbf{u}_f &= \underbrace{\begin{bmatrix}\mathbf{I}_n & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{I}_n & \cdots & \mathbf{I}_n \\ \mathbf{I}_n & \cdots & \mathbf{I}_n\end{bmatrix}}_{\mathbf{\Pi}\ (PH\,nu \times CH\,nu)} \underbrace{\begin{bmatrix}\Delta\mathbf{u}_k \\ \vdots \\ \Delta\mathbf{u}_{k+CH-1}\end{bmatrix}}_{\Delta\mathbf{u}_f} + \underbrace{\begin{bmatrix}\mathbf{I}_n \\ \vdots \\ \mathbf{I}_n\end{bmatrix}}_{\mathbf{\Phi}\ (PH\,nu \times nu)} \mathbf{u}_{k-1}, \\ \mathbf{u}_f &= \mathbf{\Pi}\Delta\mathbf{u}_f + \mathbf{\Phi}\mathbf{u}_{k-1},\end{aligned} \qquad (12)$$

Next, the proposed MPC formulation is presented. According to usual practical formulation of MPC for tracking set-points (Zanin, Tvrzská de Gouvea and Odloak 2002), the proposed controller online determines the sequence of control moves $\Delta\mathbf{u}'_f = [\Delta\mathbf{u}'_k \cdots \Delta\mathbf{u}'_{k+CH-1}]$ that minimizes a performance index -which penalizes the distance between prediction and references- subject to variable constraints. The optimization problem arising from this idea reads:

$$\begin{aligned}&\min_{\Delta\mathbf{u}_f}\left\|\mathbf{W}_{\mathbf{y}}\left(\hat{\mathbf{y}}_f - \mathbf{r}_f\right)\right\|^2 + \left\|\mathbf{W}_{\mathbf{u}}\Delta\mathbf{u}_f\right\|^2 \\ &s.t. \\ &\quad \Delta\mathbf{u}_{f,\min} \le \Delta\mathbf{u}_f \le \Delta\mathbf{u}_{f,\max} \\ &\quad \mathbf{u}_{f,\min} \le \mathbf{u}_f \le \mathbf{u}_{f,\max}\end{aligned} \qquad (13)$$

where $\mathbf{r}'_f = [\mathbf{r}'_k \ldots \mathbf{r}'_{k+PH-1}]$ is the desired near trajectory portion, $\mathbf{W}_{\mathbf{y}}$ and $\mathbf{W}_{\mathbf{u}}$ are the weighting diagonal matrices penalizing the output deviation from the trajectory and the input move, respectively. As a receding horizon policy is applied, only the first element of $\Delta\mathbf{u}_f$ (i.e., $\Delta\mathbf{u}_k$) is injected to the plant and, at the next sample time, the same optimization problem is solved again. The input constraint vectors are composed as follows:

$$\begin{aligned}\Delta\mathbf{u}'_{f,\max} &= \Delta\mathbf{u}'_{\max}\left[\mathbf{I}_{nu} \ldots \mathbf{I}_{nu}\right]_{nu \times CHnu}, \\ \Delta\mathbf{u}'_{f,\min} &= \Delta\mathbf{u}'_{\min}\left[\mathbf{I}_{nu} \ldots \mathbf{I}_{nu}\right]_{nu \times CHnu}, \\ \mathbf{u}'_{f,\max} &= \mathbf{u}'_{\max}\left[\mathbf{I}_{nu} \ldots \mathbf{I}_{nu}\right]_{nu \times PHnu}, \\ \mathbf{u}'_{f,\min} &= \mathbf{u}'_{\min}\left[\mathbf{I}_{nu} \ldots \mathbf{I}_{nu}\right]_{nu \times PHnu}.\end{aligned}$$

Now, the optimization problem will be put in a quadratic programming (QP) form. Using the prediction model (Eq. 10) and Eq. (12), and reorganizing the cost function and the constraints, the optimization problem (Eq. 13) can be written as follows:

$$\begin{aligned}&\min_{\Delta\mathbf{u}_f}\Delta\mathbf{u}'_f\mathbf{H}_k\Delta\mathbf{u}_f + 2\mathbf{f}'_k\Delta\mathbf{u}_f \\ &s.t. \\ &\underbrace{\begin{bmatrix}\mathbf{I} \\ -\mathbf{I} \\ \mathbf{\Pi} \\ -\mathbf{\Pi}\end{bmatrix}}_{\mathbf{A}}\Delta\mathbf{u}_f \le \underbrace{\begin{bmatrix}\Delta\mathbf{u}_{f,\max} \\ -\Delta\mathbf{u}_{f,\min} \\ \mathbf{u}_{f,\max} - \mathbf{\Phi}\mathbf{u}_{k-1} \\ -\mathbf{u}_{f,\min} + \mathbf{\Phi}\mathbf{u}_{k-1}\end{bmatrix}}_{\mathbf{b}_k},\end{aligned} \qquad (14)$$

where

$$\begin{aligned}\mathbf{H}_k &= \mathbf{\Pi}'D^{-1}_{\mathbf{u}_f}\mathbf{C}'_{2,k}D_{\mathbf{y}_f}\mathbf{W}^2_{\mathbf{y}}D_{\mathbf{y}_f}\mathbf{C}_{2,k}D^{-1}_{\mathbf{u}_f}\mathbf{\Pi} + \mathbf{W}^2_{\mathbf{u}}, \\ \mathbf{f}'_k &= [D_{\mathbf{y}_f}\mathbf{C}_{1,k}D^{-1}_{\mathbf{x}_{p1,k}}(\mathbf{x}_{p1,k} - \bar{\mathbf{x}}_{p1,k}) + D_{\mathbf{y}_f}\mathbf{C}_{2,k}D^{-1}_{\mathbf{u}_f}(\mathbf{\Phi}\mathbf{u}_{k-1} - \bar{\mathbf{u}}_f) \\ &\quad + \bar{\mathbf{y}}_f + \mathbf{K}(\mathbf{y}_k - \hat{\mathbf{y}}_k) - \mathbf{r}_f]'\,\mathbf{W}^2_{\mathbf{y}}D_{\mathbf{y}_f}\mathbf{C}_{2,k}D^{-1}_{\mathbf{u}_f}\mathbf{\Pi},\end{aligned}$$

where $\mathbf{x}_{p1,k}$ and $\Delta\mathbf{u}_f$ are in their original units. Notice that this optimization problem can be solved with any existing QP solver included in commercial MPC controllers (Qin and Badgwell 2003). In all cases, it should be verified that $rank\left(\mathbf{P}'_{p,k}\mathbf{P}_{p,k}\right) = A$, for all k (see Eq. 8), before implementing the proposed controller.

In order to tune the controller, the number of latent variables retained in the PCA model, *A*, (see Eq. 7) is firstly determined and coincides with the degrees of freedom of the process. Since a correlation model (with rank *A*) is used for prediction, then the problem in Eq. (13) will be well-conditioned provided that the numbers of independent sources propagated by the model is lower than or equal to *A*. Then, the control horizon is set to $CH = A/n$ in order for the number of independent sources (composed by the consecutive control moves, *CH*, times the measurement vector dimension, *n*) to be equal to the degrees of freedom *A* (i.e., $CH\,n = A$). Furthermore, the prediction horizon is set to $PH = (A - nu\,CH)/ny$, in such a way that the sum of the control and prediction horizons are equal to *A* (i.e., $nu\,CH + ny\,PH = A$).

Note that Eq. (8) represents a dynamic model with time-varying coefficients $\mathbf{C}_k$, i.e., Eq. (8) is an autoregressive with exogenous variables (ARX) model for each time *k*. This non-parsimonious model structure is better identified using latent variable methods (Duchesne and MacGregor 2001), but the identified model reliability depends of the selection of *L* and *A*. In process control, the objective of PCA model identification is not just to obtain a model which give good predictions of the controlled variables (outputs), but to obtain a good approximation to the true underlying dynamic behavior of the process so that the controller design (involving inversion of the model structure) results in good control of the controlled variables. In the last mentioned paper, it was stated that the sum of squares of the model residuals (SSE) for each retained latent variable should not be used alone to select the appropriate dynamic model. They proposed to complement the SSE profile with a model parameters uncertainty (or stability) profile which would reveal when the model is over-fitting the data. A jackknife criterion was used for measure this model parameters uncertainty for each retained latent variables. However, this criterion is not directly applicable to our case, because in this work a dynamic model (for each time) is extracted from a multi-way PCA model. Hence, in order to adapt this technique to our case, it is necessary to calibrate a PCA model for each piece of data around each time *k*, $\mathbf{X}_k^p = \mathbf{X}(1 \ldots I(L+1), k-2PH \ldots k+PH-1)$, which are associated to each dynamic model $\mathbf{C}_k$ (with $k = 2PH+1 \ldots K-L-PH$). However, this requires further analysis that is beyond the scope of this paper, and so, only the SSE is used in this work to determine *A*.

Finally, it should be noted that opposite to other existing LV-MPC formulations (Golshan, MacGregor, Bruwer and Mhaskar 2010), the proposed strategy explicitly includes input and input move constraints. This fact, not only allow us to fulfill the variable limits (which can be done by any saturation device) but to predict, and then to anticipate, the possible saturation of the variable. This results, as will be shown later, in a better use of the control inputs.

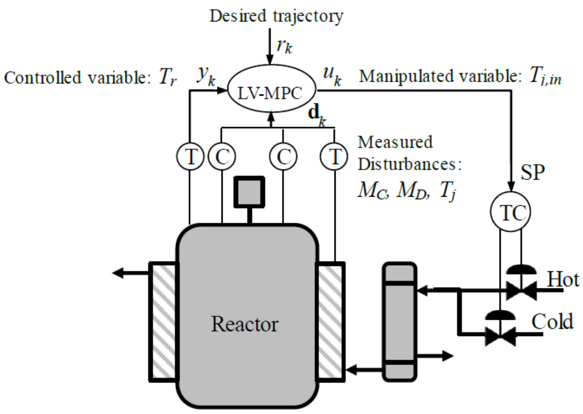

Figure 1: Schematic of the reactor with its LV-MPC instrumentation given by r_k, y_k, $\mathbf{d}_k$ and u_k. The instrumentation of the PI control (SISO) comprises only r_k, y_k and u_k.

3. CONSTRAINED LV-MPC FOR BATCH REACTOR TRACKING CONTROL

In this section, a batch reactor is used to illustrate the trajectory tracking capabilities and properties of the proposed algorithm. Aziz, Hussain and Mujtaba (2000) presented a nonlinear model of this batch reactor. This process model was originally proposed by Cott and Macchieto (1989) as a case study for a temperature control problem on a batch reactor. A complete description of the model equations is found in Appendix A, which details how the quantities of products M_C, M_D are dynamically produced from the quantities of raw materials M_A, M_B. Values for the model parameters, under nominal conditions, are the same as those reported in Aziz, Hussain and Mujtaba (2000) (see Table A.1). Fig. 1 shows the schematic of the batch reactor system. The reactor temperature (T_r) is used as the controlled variable (y_k), which is bounded between 20 and 100°C. The jacket temperature set-point ($T_{j,in}$) is used as manipulated variable (u_k) and is bounded between 20 and 120°C. The control objective is to track the reactor temperature set-point (r_k) by adjusting the inlet jacket temperature. The set-point trajectory is arbitrarily complex and it was used in Yu and Flores-Cerrillo (2013). On-line reactor temperatures, considered to be available every 0.2 min, are corrupted by normally distributed random error with standard deviation σ=0.15°C. The total batch time is 150 min and the initial values of $[M_A\, M_B\, M_C\, M_D\, T_j\, T_r]$ are $[12\ 12\ 0\ 0\ 25\ 25]$, respectively. Control action for both PI and LV-MPC (see Fig. 1) is taken every 0.2 min. 16 batch runs were collected under the same PI controller (Kc=15 and Ti =10) (Aziz, Hussain and Mujtaba 2000) with similar level of RBS excitation (1 nominal batch run and 15 batch runs with RBS excitation). A RBS dither signal is added to the manipulated variable with a normalized frequency 0.15, i.e., that the signal remains constant over 6-7 sample times on average which is suitable with respect to the process time constant. The dither magnitude around the input (±12 °C ≈ ±15% of the trajectory mean) was small enough to have little noticeable effect on the temperature trajectories. The mean absolute error for the controlled variable is only about 30% higher with the added RBS than without added RBS signal (i.e. when the system is controlled only by the PI). In order to test the consistency of the proposed design methodology, the following measurements vector is considered, $\mathbf{x}_k = [T_r \, T_{j,in} \, M_C \, M_D \, T_j]'_k = [y_k \, u_k \, \mathbf{d}'_k]'$, where n=5. A regularization parameter L=5 was adopted to construct the data matrix **X**. Wold's R criterion is used to select the number of principal components (A) retained in the PCA model (see Eq. 4), which is given by: $R(a+1) = MSE(a+1) / MSE(a)$, where $MSE(a) = SSE(a) / (I(L+1)(K-L)n)$ is the Mean Squared Error using “a” latent variables. The inclusion of new latent variables into the model finishes when the ratio $R(a+1)$ exceeds a predefined threshold of 0.9 in this case and hence A=a. Figure 2 shows the selection criterion used to set A=15, the standard error $RMSE(a)$, and the explained variance given by $R_X^2(a) = tr(\hat{\mathbf{X}}'(a)\hat{\mathbf{X}}(a)) / tr(\mathbf{X}'\mathbf{X})$, where $\hat{\mathbf{X}}(a)$ is the prediction using “a” latent variables. Conceptually, this criterion states that an additional latent variable will not be included in the model unless it provides a meaningful prediction improvement, and consequently, it gives the maximum number of latent variables to be included in the model. The control horizon is set to CH=3, because A=15 and n=5. Then, the prediction horizon is set to PH=12. The limits of the rate of change are: $\Delta u_{\max} = 15$, $\Delta u_{\min} = -15$. The following weights are used: $\mathbf{W_y} = (1/E_{Tr} PH^{0.5})\mathbf{I}_{PH}$ and $\mathbf{W_u} = s/(\Delta u_{\max} - \Delta u_{\min})\mathbf{I}_{CH}$, where $E_{Tr} = 0.35\ °C$ is the allowable mean error and $s = 3.6$ is the move suppression factor.

In order to evaluate the constrained LV-MPC algorithm, several studies are performed to investigate the effect of the information content of the data available for model building, the type of model (adaptive vs. fix), and the performance under different conditions.

Figure 3a shows the trajectory tracking of the PI controller during a nominal batch run, in order to benchmark the proposed LV-MPC methodology against the commonly used controller. The PI controller is tightly tuned, in contrast to the PI controller that was used in Golshan, MacGregor, Bruwer and Mhaskar (2010) (evidenced by its poor performance in the simulations). Figure 4 shows the trajectory tracking of the constrained LV-MPC during a nominal batch run.

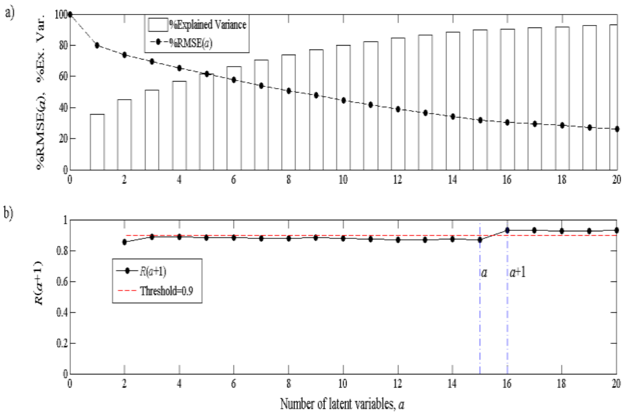

Figure 2: Selection of the number of latent variables by using the criterion $R_{0.9}$. a) Pareto of explained variability and percentage of standard error. b) Successive errors rate together with its threshold for selection of the model order (A=15).

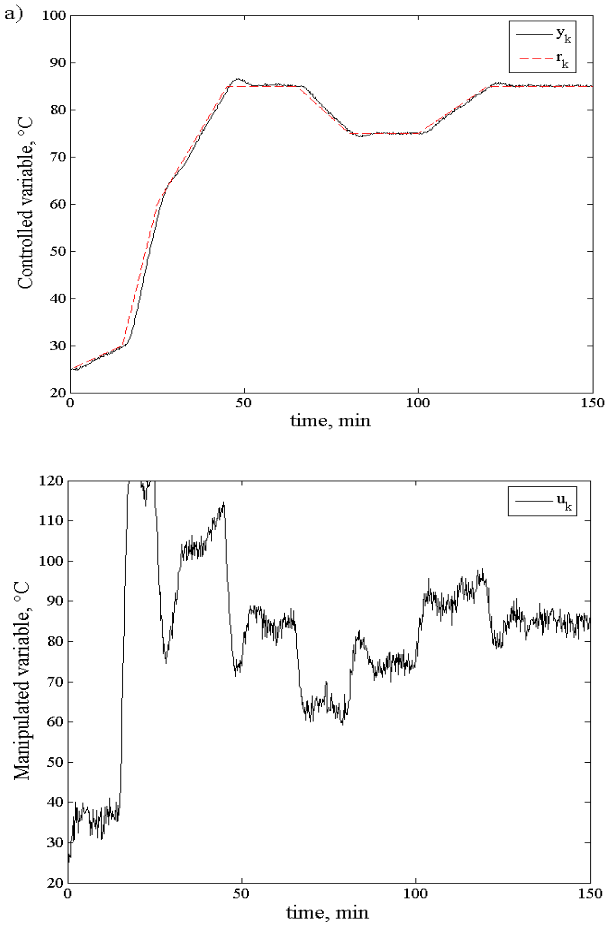

Figure 3: Implementation of a tightly tuned PI controller for the temperature control problem.

In Golshan, MacGregor, Bruwer and Mhaskar (2010), the desired set-point trajectory is included in the modeled data and in the prediction model. If this trajectory varies continuously (as it was in their case study), the numerical conditioning (rank of **P**) is increased by improving the predictive power. In this paper, the trajectory is not included in the modeled data. Even so, a very good control is obtained (see Fig. 4).

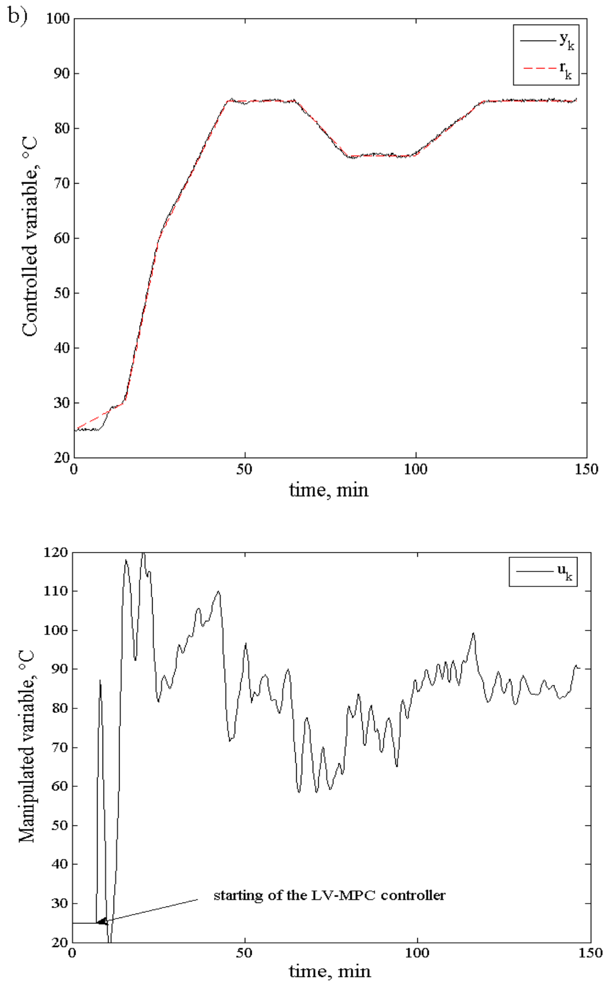

Figure 4: Implementation of constrained LV-MPC using a PCA model based on RBWU dataset.

The proposed tuning methodology produces a LV-MPC with tighter tracking using a more parsimonious model, which was calibrated using less batch runs than in Golshan, MacGregor and Mhaskar (2011). It is also important to check the power of offset elimination and disturbance rejection for the proposed control methodology, in particular, the ability of the controller to incorporate integral action to reject the effect of non-stationary disturbances. The set-point tracking study shows no evident offsets even when the set-point trajectories are a sequence of ramps (see Fig. 4). This is because the models are based on the variable deviations about the mean trajectories and the control moves $\Delta\mathbf{u}_k$ -calculated by the LV-MPC- are then added to $\mathbf{u}_{k-1}$ to get the final $\mathbf{u}_k$ setting. In the LV-MPC methodology proposed here, offset is handled automatically by the offset correction (see Eq. 10) and by the information on the non-stationary effects of the disturbances (that are built into the PCA model developed from the training data). In Golshan, MacGregor, Bruwer and Mhaskar (2010) and Golshan, MacGregor and Mhaskar (2011), the manipulated variable takes negative values (i.e., they are physically unrealizable). In contrast, the proposed controller effectively manages the constraints of value and rate of change in the manipulated variable. To provide a severe test of the disturbance rejection ability of the batch constrained LV-MPC, a large additional random walk disturbance was added to the measured reactor temperature (i.e., the controlled variable) for several simulation runs and the ability of the LV-MPC to eliminate the large offsets coming from this disturbance was investigated. The study is intended only as a severe test of the ability of the LV-MPC to eliminate offset due to non-stationary disturbances. Figure 5 shows the random walk disturbance that was added directly to the controlled variable. If there was no offset elimination (integral action) then the 10–20°C offsets would appear in the controller's tracking of the set-point trajectory.

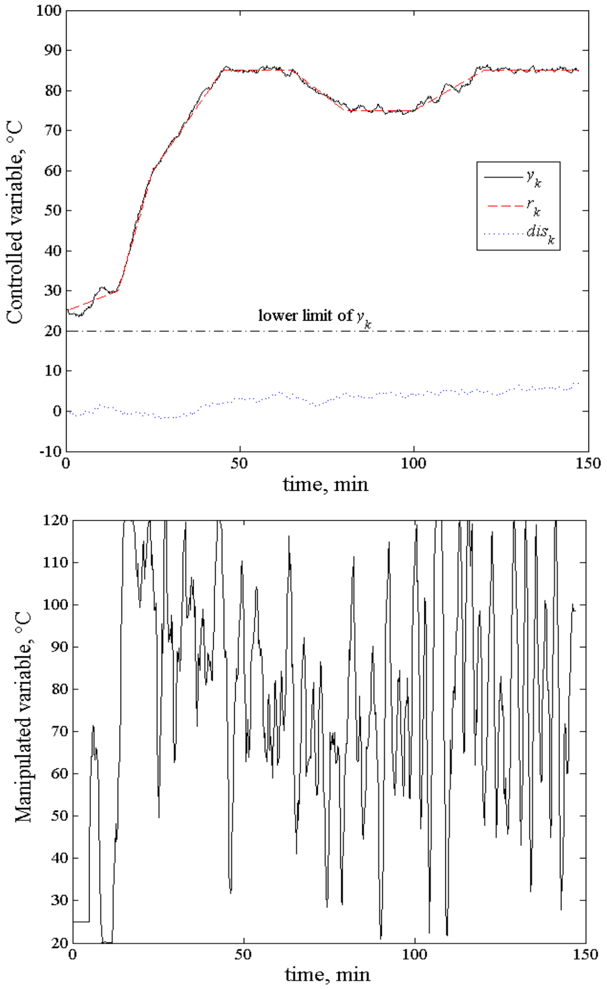

Figure 5. Performance of LV-MPC for both tracking the set-point trajectory and rejecting a random walk disturbance occurring on top of the output (T_r).

However, the proposed control methodology is clearly able to reject the non-stationary disturbances. The input moves in Fig. 5 are more aggressive than in Fig 4, mainly due to the severity of the test disturbance, although the input constraints are still satisfied.

In summary, the results of the simulations show that the proposed technique is highly efficient. A tighter tracking with a smoother input (fulfilling constraints), along with a high performance disturbance rejection promote the industrial application of the proposed technique.

4. CONCLUSIONS

This paper presents a constrained LV-MPC technique for trajectory tracking in batch process. Our approach consists in building a MPCA model applicable to every time point based on a moving window along the batch-wise unfolded dataset. The proposed control and tuning technique is quite simple and robust, which makes it attractive to the biotechnology and pharmaceutical industries where the cost of data collection is very important. Furthermore, most commercial MPC software has QP solvers, and hence, the offset-free LV-MPC formulation could be easily implemented. A case study has been presented to illustrate the proposed technique and to show the good performance of the controller.

In a future work it is intended to study the following points: (i) the development of a reliable method to compute the number of shifts in matrix **X**, *L*, and the number of latent variables retained in the model, *A*; (ii) the need to change the local correlation model at each sample time along the batch; and (iii) the extension of the current LV-MPC formulation to include economic objectives.

ACKNOWLEDGMENTS

The authors are grateful for the financial support received from CONICET.

APPENDIX A. PROCESS MODEL FOR THE CASE STUDY

The model equations for the batch reactor are as follows:

$$k_1 = \exp\left(k_1^1 - \frac{k_1^2}{T_r + 273.15} \right) \quad \text{(A.1)}$$

$$k_2 = \exp\left(k_2^1 - \frac{k_2^2}{T_r + 273.15} \right) \quad \text{(A.2)}$$

$$R_1 = k_1 M_A M_B \quad \text{(A.3)}$$

$$R_2 = k_2 M_A M_C \quad \text{(A.4)}$$

$$Q_r = -\Delta H_1 R_1 - \Delta H_2 R_2 \quad \text{(A.5)}$$

$$M_r = M_A + M_B + M_C + M_D \quad \text{(A.6)}$$

$$C_{pr} = \frac{C_{pA} M_A + C_{pB} M_B + C_{pC} M_C + C_{pD} M_D}{M_r} \quad \text{(A.7)}$$

$$Q_j = UA(T_j - T_r) \quad \text{(A.8)}$$

$$\frac{dM_A}{dt} = -R_1 - R_2 \quad \text{(A.9)}$$

$$\frac{dM_B}{dt} = -R_1 \quad \text{(A.10)}$$

$$\frac{dM_C}{dt} = +R_1 - R_2 \quad \text{(A.11)}$$

$$\frac{dM_D}{dt} = +R_2 \quad \text{(A.12)}$$

$$\frac{dT_r}{dt} = \frac{(Q_r + Q_j)}{M_r C_{pr}} \quad \text{(A.13)}$$

$$\frac{dT_j}{dt} = \frac{(T_j^{SP} - T_j)}{\tau_j} - \frac{Q_j}{V_j \rho_j C_{pj}} \quad \text{(A.14)}$$

The above model parameters are given in Table A.1.

Table A.1. Constant parameter in the reactor model.

Parameter	Value
C_{pA}	18.0 kcal/kmol °C
C_{pB}	40.0 kcal/kmol °C
C_{pC}	52.0 kcal/kmol °C
C_{pD}	80.0 kcal/kmol °C
ΔH_1	-10000.0 kcal/kmol
ΔH_2	-6000.0 kcal/kmol
C_{pj}	0.45 kcal/kg °C
U	9.76 kcal/min m^2 °C
ρ_j	1000.0 kg/m^3
A	6.24 m^2
V_j	0.6921 m^3
k_1^1	20.9057
k_1^2	10000
k_2^1	38.9057
k_2^2	17000
Δt	0.2 min
τ_j	3 min

REFERENCES

Angeli D., Amrit R., Rawlings J.B., 2012. On average performance and stability of economic model predictive control, IEEE Trans. Automatic Control, 57, 1615–1626.

Alamo T., Ferramosca A., González A.H., Limon D., Odloak D., 2014. A gradient-based strategy for the one-layer RTO+MPC controller. J. Process Control, 24, 435–447.

Altintena A., Ketevanlioğlua F., Erdoğana S., Hapoğlub H., Alpbaz M., 2008. Self-tuning PID control of jacketed batch polystyrene reactor using genetic algorithm. Chem. Eng. J., 38, 490−497.

Aziz N., Hussain M.A., Mujtaba I.M., 2000. Performance of different types of controllers in tracking optimal temperature profiles in batch reactors. Comput. Chem. Eng. 24, 1069–1075.

Bishop C, 1995. Neural Networks for Pattern Recognition; Clarendon Press: Oxford.

Camacho E.F., Bordons C., 2004. Model Predictive Control, 2nd ed., Springer-Verlag.

Clarke-Pringle T., MacGregor J.F., 1997. Nonlinear adaptive temperature control of multi-product, semi-batch polymerization reactors. Comput. Chem. Eng., 21, 1395−1409.

Cott B.J., Macchietto S., 1989. Temperature control of exothermic batch reactors using generic model control. Ind. Eng. Chem. Res. 28, 1177–1184.

Duchesne C., MacGregor J.F., 2001. Jackknife and bootstrap methods in the identification of dynamic models. J. Process Control, 11, 553-564.

Diehl M., Amrit R., Rawlings J.B., 2011. A Lyapunov function for Economic Optimizing Model Predictive Control, IEEE Trans. Automatic Control, 56, 703–707.

Ferramosca A., González A.H., Limon D., Bustos G.A., Godoy J.L., Marchetti J.L., 2014. On Economic Optimality of Model Predictive Control. IEEE Latin America Trans., 12, 1234-1241.

Flores-Cerrillo J., MacGregor J.F., 2005. Latent Variable MPC for trajectory tracking in batch processes. J. Process Control, 15, 651–663.

Ferramosca, A., Limon D., González A.H., Alvarado I., Camacho E.F., 2012. Robust MPC for tracking zone regions based on nominal predictions. J. Process Control, 22, 1966–1974.

García C.E., 1984. Quadratic Dynamic Matrix Control of Nonlinear Processes. An Application to a Batch Reaction Process. AIChE Annual Meeting, San Franciso, CA.

Gattu G., Zafiriou E., 1992. Nonlinear dynamic matrix control with state estimation. Ind. Eng. Chem. Res., 31, 1096−1104.

Golshan M., MacGregor J.F., Bruwer M.J., Mhaskar P., 2010. Latent Variable Model Predictive Control (LV-MPC) for trajectory tracking in batch processes. J. Process Control 20, 538–550.

Golshan M., MacGregor J.F., Mhaskar P, 2011. Latent variable model predictive control for trajectory tracking in batch processes: Alternative modeling approaches. J. Process Control, 21, 1345–1358.

Godoy J.L., Vega J.R., Marchetti J.L., 2014. Relationships between PCA and PLS-regression. Chem. Intell. Lab. Syst., 130, 182–191.

González A.H., Adam E.J., Marchetti J.L., 2008. Conditions for offset elimination in state space receding horizon controllers: A tutorial analysis. Chemical Engineering and Processing 47, 2184–2194.

Kravaris C., Wright R.A., Carrier J.F., 1989. Nonlinear controllers for trajectory tracking in batch processes. Comput. Chem. Eng. 13, 73−82.

Kravaris C., Saroush M., 1990. Synthesis of multivariate nonlinear controllers by input/output linearization. AIChE J., 36, 249−264.

Lontra H.C., 1991. Modelagem e Simulação de um Reator Batelada de Polimerização de Estireno em Suspensão. Anais do Congresso Brasileiro de Polimeros Saõ Paulo, Brazil, 619−623.

MacGregor J.F., Bruwer M.J., Golshan M., 2009. System and methodology for the model predictive control of batch processes using latent variable dynamic models, U.S. Patent No. : US 2009/0287320 A1.

Martin E., Morris A., 1996. Non-parametric Confidence Bounds for Process Performance Monitoring Charts. J. Process Control, 6, 349-358.

Mujtaba I.M., Aziz N., Hussain M.A., 2006. Neural network based modeling and control in batch reactor. Chem. Eng. Res. Des., 84, 635–644.

Nomikos P., MacGregor J.F., 1994. Monitoring batch processes using multiway principal component analysis. AIChE J., 40, 1361–1375.

Nelson P.P.C, Taylor P.A., MacGregor J.F., 1996. Missing data methods in PCA and PLS: score calculations with incomplete observations. Chemom. Intell. Lab. Syst., 35, 45–65.

Peterson T., Hernandez E., Arkun Y., Schork J.F., 1992. A nonlinear DMC algorithm and its application to a semi-batch polymerization reactor. Chem. Eng. Sci., 47, 737−753.

Qin S.J., Badgwell T.A., 2003. A survey of industrial model predictive control technology, Control Eng. Pract., 11, 733–764.

Rawlings J.B., Mayne D.Q., 2009. Model Predictive Control: Theory and Design, Nob-Hill Publishing, 1st. edition.

Troup G.M., Georgakis C., 2013. Process systems engineering tools in the pharmaceutical industry. Comput. Chem. Eng., 51, 157– 171.

Wang Z.L., Pla F., Corriou J.P., 1995. Nonlinear adaptive control of batch styrene polymerization. Chem. Eng. Sci., 50, 2081−2091.

Yu H., Flores-Cerrillo J., 2013. Latent Variable Model Predictive Control for Trajectory Tracking in Batch Processes: Internal Model Control Interpretation and Design Methodology. Ind. Eng. Chem. Res., 52, 12437−12450.

Zanin A.C., Tvrzská de Gouvea M., Odloak D., 2002. Integrating real time optimization into the Model Predictive Controller of the FCC system. Control Eng. Practice, 10, 819–831.