Journal of Process Control 45 (2016) 1–11

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Constrained latent variable model predictive control for trajectory tracking and economic optimization in batch processes

J.L. Godoy[a,b,*], A.H. González[b], J.E. Normey-Rico[a]

[a] Department of Automation and Systems (DAS), Universidade Federal de Santa Catarina (UFSC)—CTC, CEP: 88040-900 Florianopolis, Brazil
[b] INTEC (CONICET and Universidad Nacional del Litoral), Güemes 3450, 3000 Santa Fe, Argentina

ARTICLE INFO

Article history:
Received 2 December 2015
Received in revised form 24 May 2016
Accepted 15 June 2016

Keywords:
Multi-way principal component analysis
Latent variable model
Constrained predictive control
Economic cost function
Model validity indicator

ABSTRACT

A constrained latent variable model predictive control (LV-MPC) technique is proposed for trajectory tracking and economic optimization 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 rearranged by means of a regularized batch-wise unfolding. The main advantages of LV-MPC over other MPC techniques are: (i) requirements for the dataset are rather modest (only around 10–20 batch runs are necessary), (ii) nonlinear processes can efficiently be handled algebraically through MPCA models, and (iii) the tuning procedure is simple. The LV-MPC for tracking is tested through a benchmark process used in previous LV-MPC formulations. The extension to economic LV-MPC includes an economic cost and it is based on model and trajectory updating from batch to batch to drive the process to the economic optimal region. A data-driven model validity indicator is used to ensure the prediction's validity while the economic cost drives the process to regions with higher profit. This technique is validated through simulations in a case study.

1. Introduction

The phenomenological inner behavior of most of the batch pharmaceutical processes are not well understood, and so, fundamental – or even knowledge driven – models are difficult to obtain. An additional roadblock in the development of such models is the small production rates of the majority of pharmaceutical products compared to the production rate of bulk chemical and petrochemicals for which a plethora of knowledge-driven models has found extensive use over the last four to six decades. For the majority of pharmaceutical processes, or their processing steps, one needs to rely substantially on the development of data-driven models. Particularly, Troup and Georgakis [1] have motivated the need of data-driven rather than knowledge-driven models, given that they are suitable for quick deployment in process optimization and online control tasks related to pharmaceutical processes.

It is well known that batch and semi-batch processes are used in many industries because of their flexibility to manage many different grades and types of products. In these processes, one of the requirements to achieve consistent final quality specifications and adequate operation is to track references signals determined by optimization layers. 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 not achieve adequate control performance. Enhancements to conventional PID controllers have proven to lessen some of these deficiencies: PID-feed forward controllers [2], adaptive PID controllers [3], and self-tuning PID controllers [4]. 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 [5,6] and generic model control [7–9].

Another advanced control strategy used for these processes is Model Predictive Control (MPC). Garcia [10] implemented a MPC strategy for the temperature control of synthetic rubber production in a semi-batch process; Gattu and Zafiriou [11] extended the work of Garcia by incorporating Kalman filter estimation. Multi-input multi-output (MIMO) MPC was addressed by Peterson et al. [12] 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 ones), taking into account a dynamic simplified model of the plant, constraints, and stability require-

* Corresponding author at: INTEC (CONICET and Universidad Nacional del Litoral), Güemes 3450, 3000 Santa Fe, Argentina.
E-mail address: jlgodoy@santafe-conicet.gov.ar (J.L. Godoy).

http://dx.doi.org/10.1016/j.jprocont.2016.06.005
0959-1524/

ments [13,14]. That is why MPC is the most used advanced control strategy in industry.

Initially MPC industrial applications considered the solution of the dynamic and economic problem in a two-layer optimization problem. However, in the last years, the tendency of including economic objectives in the MPC controller itself has been disseminated. Several economic formulations of MPC have been presented, which get over the standard set-point-tracking formulation [15]. 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 [16]. That is why, the process control variables attempt to minimize the (possibly non-linear) economic cost [16].

The main goal of the MPC software suppliers is to provide easy tuning algorithms that require a modest experimental design (for identification) in order to easily disseminate 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 [1] highlight the interest in LV-MPC, in particular for batch processes, because it is an alternative to Non-linear 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 [17], where the prediction of the future trajectories is accomplished using statistical latent variable missing data imputation methods [18]. Flores-Cerrillo and MacGregor [19] have developed a version of LV-MPC for batch processes using a PCA model and [20] propose a design methodology to select the parameters of this LV-MPC controller. Another LV-MPC technique based on multiphase modeling of a Batch-Wise Unfolding (BWU) of data arrays has been proposed in Refs. [21,22], 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 motivates Regularized Batch-Wise Unfolding (RBWU) to decrease the number of collected batches necessary to identify a model [23]. 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 et al. [22] have shown that controlling in the latent variable space and in the manipulated variable space have the same performance. However, controlling in the manipulated variable space is preferable because the optimization variables can be directly constrained. That is why 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. This allows performing an economic extension of constrained LV-MPC. 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 (as [21–23]).

The rest of the work is organized as follows. In the next section, the constrained LV-MPC for batch processes is presented. Sections 3 and 4 are devoted to two cases studies. First, the method is tested on a simulated batch reactor, which serves as a benchmark because it has been previously used to evaluate LV-MPC techniques. The second case study considers the economic extension of constrained LV-MPC. 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. Then, an extension to economic LV-MPC is proposed which is based on (batch-to-batch) trajectory and model updating supervised by a model validity indicator to drive the process to the economic optimum without extrapolating.

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:

$$\boldsymbol{x}'_{i,k} = \left[\mathbf{y}'_{i,k}\ \boldsymbol{u}'_{i,k}\ \boldsymbol{d}'_{i,k}\right] \in \Re^{1\times n}, \tag{1}$$

where $\mathbf{y}_{i,k} \in \Re^{ny}$, $\mathbf{u}_{i,k} \in \Re^{nu}$ and $\mathbf{d}_{i,k} \in \Re^{nd}$ are the controlled, manipulated, and measured variables, respectively; and $n = ny + nu + nd$. $\mathbf{d}_{i,k}$ is a vector of on-line measurements (e.g., pressure, temperature, stirring, flows, 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 [23] stems from the strategies utilized to construct a 2-dimensional array (a matrix) from the 3-dimensional data cube. Nomikos and MacGregor [17] suggested that the batch-wise unfolding (BWU) approach is the most logical way for modeling the differences among batches. In the proposed 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.,:

$$\boldsymbol{X}_B = [\mathbf{X}_1 \cdots \boldsymbol{X}_k \cdots \boldsymbol{X}_K] \in \Re^{I\times Kn}, \text{where} \boldsymbol{X}_k = \begin{bmatrix} \mathbf{x}'_{1,k} \\ \vdots \\ \boldsymbol{x}'_{I,k} \end{bmatrix} \in \Re^{I\times n}, \tag{2}$$

In the BWU, 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 nonlinearity characteristics of the batch. The calibration data can be the data obtained from the previous batches run in normal conditions, augmented with additional batches executed according to identification experiments. This is so as to provide more information on the causal relationships at every time interval.

In this work, regularized batch-wise unfolding (RBWU) is used because it produces a regularized version of the PCA model of $\mathbf{X}_B$ and requires less batch runs [23]. This approach unfolds batch-wise but also repeats each batch row L times, each time shifted by one additional sampling interval, as follows:

$$\boldsymbol{X} = \begin{bmatrix} \mathbf{X}_1 & \boldsymbol{X}_2 & \cdots & \boldsymbol{X}_{K-L} \\ \boldsymbol{X}_2 & \boldsymbol{X}_3 & \cdots & \boldsymbol{X}_{K-L+1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{X}_{L+1} & \mathbf{X}_{L+2} & \cdots & \mathbf{X}_K \end{bmatrix} \in \Re^{I(L+1)\times(K-L)n}, \tag{3}$$

The parameter L is the number of shifts in matrix $\mathbf{X}$. If $L=0$ (no shift), the unfolding is simply BWU (Eq. (2)). But if a small number of shifts (e.g., $L/K < 0.05$) is used, this approach will retain most of the advantages of the BWU approach (by capturing time varying non-linear behavior), but 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. This modeling aims at capturing the major benefits of the two previously mentioned modeling approaches, while avoiding the problems related to each one [23]. A PCA model based on the unfolded matrix $\mathbf{X}$ (Eq. (3)) that contains mean-centered and scaled data produces a reduced dimension latent model, as follows:

$$\hat{\mathbf{X}} = \mathbf{X}\mathbf{P}\mathbf{P}', \tag{4}$$

where $\mathbf{P} \in \Re^{(K-L)n \times A}$ is the loading matrix, A being the number of retained latent variables [24]. The value of A must be smaller than or equal to the number of batches I available in the calibration dataset (i.e., $A \leq I$), for adequately modeling. 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 instantaneous loadings (i.e., at each time), thereby modeling the instantaneous covariance structure and its changes over 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 $\mathbf{X}$ is the average of the covariance matrix of batch-wise unfolded matrix $\mathbf{X}_B$ over $(L+1)$ consecutive sample times [23]. Thus, the resulting covariance matrix is a regularized version of the original covariance matrix. Note that the loading matrix $\mathbf{P}$ of the PCA model built for the $\mathbf{X}$ 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 executed 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 of 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 has previously been used in Ref. [22]. 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 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 $\mathbf{X}$ 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:

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

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

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

where coefficients in $\mathbf{P}'_{p1,k}$ and $\mathbf{P}'_{f1,k}$ account for the correlations between measurements, from k-2PH to $k+PH$-1.

Then, the near future process variables $\mathbf{x}_{f1,k}$ can be estimated using missing data imputations, as follows:

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

where $rank(\mathbf{P}'_{p1,k}\boldsymbol{P}_{p1,k}) = rank(\mathbf{P}'\mathbf{P}) = A$. This prediction is based on the following latent relationship: $[\mathbf{x}'_{f1,k}, \mathbf{x}'_{p1,k}] = \mathbf{t}'_k[\mathbf{P}'_{f1,k}, \mathbf{P}'_{p1,k}]$, where the latent variables, $\mathbf{t}_k$, could be estimated by using the known part of the data, i.e., $\hat{\mathbf{t}}_k = (\mathbf{P}'_{p1,k}\mathbf{P}_{p1,k})^{-1}\mathbf{P}'_{p1,k}\mathbf{x}_{p1,k}$; and hence, $\hat{\mathbf{x}}_{f1,k} = \mathbf{P}_{f1,k}\hat{\mathbf{t}}_k$ (see Godoy et al. [24] for better understanding). 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 (see Eq. (1)) improves the consistency of the correlation model (Eq. (4)) and therefore the predictive model (Eq. (7)).

Remark 1. 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 $\mathbf{x}$. 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 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 Ref. [22], 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, $\mathbf{y}'_f = [\mathbf{y}'_k \cdots \mathbf{y}'_{k+PH-1}]$, will be predicted by means of the recent past data, $\mathbf{x}_{p1,k}$, and future control inputs, $\mathbf{u}'_f = [\mathbf{u}'_k \cdots \mathbf{u}'_{k+PH-1}]$, 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 $\mathbf{x}'_{p,k} = [\mathbf{x}'_{p1,k}, \mathbf{u}'_f]$ is first defined, and then, the appropriate model partition corresponding to the this vector, $\boldsymbol{P}'_{p,k} = [\boldsymbol{P}'_{p1,k} \quad \boldsymbol{P}'_{u,k}]$, is computed. Matrix $\mathbf{P}_{u,k} \in \Re^{PHnu\times A}$, in the later partition, contains the rows of $\mathbf{P}_{f1,k} \in \Re^{PHn\times A}$ corresponding to $\mathbf{u}_f$. This way, the near future outputs can be predicted by:

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

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 $k\cdots k+PH-1k\cdots k+PH-1$. Finally, as usual in MPC closed-loop, a correction term is added to the prediction, to account for the feedback:

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

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

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

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 constitute an appropriate practice in MPC to obtain an offset-free formulation [25]. Assuming a control horizon CH, $\mathbf{u}_f$ can be written as follows:

$$\mathbf{u}_f = \underbrace{\begin{bmatrix}\mathbf{I}_n & \mathbf{0} & \mathbf{0}\\ \vdots & \ddots & \mathbf{0}\\ \mathbf{I}_n & \cdots & \mathbf{I}_n\\ \mathbf{I}_n & \cdots & \mathbf{I}_n\end{bmatrix}}_{\mathbf{\Pi}\in\Re^{PHnu\times CHnu}}\underbrace{\begin{bmatrix}\Delta\mathbf{u}_k\\ \vdots\\ \Delta\mathbf{u}_{k+CH-1}\end{bmatrix}}_{\Delta\mathbf{u}_f}+\underbrace{\begin{bmatrix}\mathbf{I}_n\\ \vdots\\ \mathbf{I}_n\end{bmatrix}}_{\mathbf{\Phi}\in\Re^{PHnu\times nu}}\mathbf{u}_{k-1}, \tag{12}$$

$$\mathbf{u}_f = \mathbf{\Pi}\Delta\mathbf{u}_f + \mathbf{\Phi}\mathbf{u}_{k-1},$$

Next, the proposed MPC formulation is presented. According to usual practical formulation of MPC for tracking set points [16], the proposed controller determines online 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:

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

where $\mathbf{r}'_f = [\mathbf{r}'_k\ldots\mathbf{r}'_{k+PH-1}]$ is the desired near trajectory portion, $\mathbf{W}_y$ and $\mathbf{W}_u$ 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: $\Delta\mathbf{u}'_{f,\max} = \Delta\mathbf{u}'_{\max}[\mathbf{I}_{nu}\ldots\mathbf{I}_{nu}] \in \Re^{nu\times CHnu}$, $\Delta\mathbf{u}'_{f,\min} = \Delta\mathbf{u}'_{\min}[\mathbf{I}_{nu}\ldots\mathbf{I}_{nu}] \in \Re^{nu\times CHnu}$, $\mathbf{u}'_{f,\max} = \mathbf{u}'_{\max}[\mathbf{I}_{nu}\ldots\mathbf{I}_{nu}] \in \Re^{nu\times PHnu}$, $\mathbf{u}'_{f,\min} = \mathbf{u}'_{\min}[\mathbf{I}_{nu}\ldots\mathbf{I}_{nu}] \in \Re^{nu\times PHnu}$.

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:

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

where

$$\boldsymbol{H}_k = \boldsymbol{\Pi}'D^{-1}_{\mathbf{u}_f}\mathbf{C}'_{2,k}D_{\mathbf{y}_f}\boldsymbol{W}^2_{\mathbf{y}}D_{\mathbf{y}_f}\mathbf{C}_{2,k}D^{-1}_{\mathbf{u}_f}\Pi+\boldsymbol{W}^2_{\mathbf{u}},$$

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

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 [26]. In all cases, it should be verified that $rank(\mathbf{P}'_{p,k}\mathbf{P}_{p,k}) = A$, 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., $CH\quad n = A$). Furthermore, the prediction horizon is set to $PH = (A - nu\quad CH)/ny$, in such a way that the sum of the control and prediction horizons are equal to A (i.e., $nuCH + nyPH = A$).

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 [27], 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 Ref. [27] 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 [27]. 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 will be 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 [22], 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.

2.5. Economic extension of constrained LV-MPC: economic LV-MPC

This section is devoted to extend the formulation provided in Section 2.4, in order to include some economic criteria into the optimization cost function. The so called economic MPC, is a MPC formulation that includes economic costs into the MPC objective function, and this can be done in two main ways: by including the economic cost as a stage cost [28,29] or by including an extra stationary terminal economic cost [30,31]. In the first case, the resulting controller optimizes the transient behavior of the closed-loop, while in the later one it computes and reaches a stationary economic optimal point.

In our context, however, the control scenario is quite different. First, we have a trajectory tracking problem instead of a stationary point tracking problem. Second, the given trajectory to be tracked is not necessarily the economic optimal one, and so it is possible to look for a better one. An important point here is that given that the trajectory is not explicitly included in the prediction model (Eq. (10)), it is possible to include an economic stage cost that looks for a better trajectory from an economic point of view (at least from batch to batch). This is not the case in other LV-MPC formulations [22]. In this context, it should be noted that the trajectory tracking objective and the economic objective could be competing objectives in the early batch runs. The idea, however, is that from batch to batch the conflict between the tracking and the economic objective is reduced by means of the trajectory and the model updating.

The proposed economic LV-MPC control law (i.e., $\Delta\mathbf{u}_k$) is derived from the solution of the following optimization problem:

$$\begin{aligned}
&\min_{\Delta\mathbf{u}_f} (1-\lambda)\ell_{track}(\hat{\mathbf{y}}_f, \mathbf{r}_f, \Delta\mathbf{u}_f) + \lambda\ell_{eco}(\hat{\mathbf{y}}_f, \mathbf{u}_f, \hat{\mathbf{d}}_f)\\
&s.t.\\
&\Delta\mathbf{u}_{f,\min} \le \Delta\mathbf{u}_f \le \Delta\mathbf{u}_{f,\max} \qquad (15)\\
&\mathbf{u}_{f,\min} \le \mathbf{\Pi}\Delta\mathbf{u}_f + \mathbf{\Phi}\mathbf{u}_{k-1} \le \mathbf{u}_{f,\max}\\
&\mathbf{y}_{f,\min} \le \hat{\mathbf{y}}_f(\mathbf{u}_{k-1}, \Delta\mathbf{u}_f) \le \mathbf{y}_{f,\max}
\end{aligned}$$

where $\ell_{track}(\hat{\mathbf{y}}_f, \mathbf{r}_f, \Delta\mathbf{u}_f)$ is the tracking cost function given at Eq. (13) and $\ell_{eco}(\hat{\mathbf{y}}_f, \mathbf{u}_f, \hat{\mathbf{d}}_f)$ is the economic cost function (or economic performance measure), that also explicitly includes the near future predictions $\hat{\mathbf{d}}_f$ of measured variables. These predictions are computed as follows: $\hat{\mathbf{d}}_f = \mathbf{P}_{d,k}(\mathbf{P}'_{p,k}\mathbf{P}_{p,k})^{-1}\mathbf{P}'_{p,k}\mathbf{x}_{p,k}$, where $\mathbf{P}_{d,k}$ is composed by the rows of $\mathbf{P}_{f1,k}$ corresponding to $\mathbf{d}_f$. Parameter λ is devoted to make a balance between the tracking and the economic objective. This formulation also added constraints on the outputs, given by: $\mathbf{y}'_{f,\max} = \mathbf{y}'_{\max}[\mathbf{I}_{ny} \ldots \mathbf{I}_{ny}] \in \Re^{ny \times PHny}$ and $\mathbf{y}'_{f,\min} = \mathbf{y}'_{\min}[\mathbf{I}_{ny} \ldots \mathbf{I}_{ny}] \in \Re^{ny \times PHny}$.

The proposed quadratic economic objective function is given by

$$\begin{aligned}
\ell_{eco}(\hat{\mathbf{y}}_f, \boldsymbol{u}_f, \hat{\mathbf{d}}_f) &= \|\mathbf{y}_{f,opt} - \hat{\mathbf{y}}_f\|^2_{\boldsymbol{W}y}\\
&+\|\mathbf{u}_{f,opt} - \boldsymbol{u}_f\|^2_{\boldsymbol{W}u} + \|\mathbf{d}_{f,opt} - \hat{\mathbf{d}}_f\|^2_{\boldsymbol{W}d},
\end{aligned} \qquad (16)$$

where $\mathbf{y}'_{f,opt} = [\mathbf{y}'_{opt} \ldots \mathbf{y}'_{opt}]$, $\mathbf{u}'_{f,opt} = [\mathbf{u}'_{opt} \ldots \mathbf{u}'_{opt}]$ and $\mathbf{d}'_{f,opt} = [\mathbf{d}'_{opt} \ldots \mathbf{d}'_{opt}]$ are the values of controlled, manipulated and measured variables corresponding to the achievable optimal production, and $\mathbf{W}y$, $\mathbf{W}u$ and $\mathbf{W}d$ are the corresponding weighting diagonal matrices. $\mathbf{y}_{opt}$, $\mathbf{u}_{opt}$ and $\mathbf{d}_{opt}$ – which are parameters for the cost (Eq. (16)) – can be determined according to process knowledge (previous operations), or even, they can be computed by solving an optimization problem of the form:

$$\begin{aligned}
&\boldsymbol{y}_{opt}, \boldsymbol{u}_{opt}, \boldsymbol{d}_{opt} = arg\ \min_{\boldsymbol{y},\boldsymbol{u},\boldsymbol{d}} \boldsymbol{f}'_y\boldsymbol{y} - \boldsymbol{f}'_u\boldsymbol{u} + \boldsymbol{f}'_d\boldsymbol{d}\\
&s.t.\\
&\quad \boldsymbol{u}_{\min} \le \boldsymbol{u} \le \boldsymbol{u}_{\max}\\
&\quad \boldsymbol{y}_{\min} \le \boldsymbol{y} \le \boldsymbol{y}_{\max}\\
&\quad \boldsymbol{d}_{\min} \le \boldsymbol{d} \le \boldsymbol{d}_{\max}
\end{aligned},$$

where $\mathbf{f}'_y\mathbf{y}_f - \mathbf{f}'_u\mathbf{u}_f + \mathbf{f}'_d\mathbf{d}_f$ represents the profit, $\mathbf{f}_y$ and$\mathbf{f}_d$ are the prices corresponding to the controlled and the measured variables, $\mathbf{f}_u$is the costs corresponding to the manipulated variables, and $\mathbf{y}$, $\mathbf{d}$ are models that relates the controlled and measured variables with the manipulated variables $\mathbf{u}$.

Notice that the economic cost $\lambda\ell_{eco}$ in Eq. (15) could move the solution to a region where the MPCA model was not calibrated (i.e., a region far from the given trajectory). To avoid this fact, which results in an excessive extrapolation from the MPCA model, it is necessary to measure on line the model validity. Next, a validity indicator for predictions is proposed.

Remark 2. The optimization problem (Eq. (15)) includes both, input and output constraints. So, feasibility problems could arise. To avoid this potential formulation drawback, it is possible to include slack variables $\boldsymbol{\varepsilon}_{\min}$ and $\boldsymbol{\varepsilon}_{\max}$ into the output constraint, as follows:

$$\boldsymbol{y}_{f,\min} - \boldsymbol{\varepsilon}_{\min} \le \hat{\mathbf{y}}_f(\boldsymbol{u}_{k-1}, \Delta\mathbf{u}_f) \le \boldsymbol{y}_{f,\max} + \boldsymbol{\varepsilon}_{\max}.$$

This way, if a quadratic penalization of the slack variable is included in the optimization cost function of Eq. (15), the slack variables will decrease iteratively, up to a null value, if possible. In this case, however, the tracking and economic objective could compete with the minimization of the slacks; i.e., it could happens that a smaller economic or tracking cost is obtained with a non-null values of the slacks than with null ones. A better solution, according to most industrial optimizing controllers (as QP-DMC, etc.), is to solve a previous optimization problem to compute the minimal slack variables necessary to obtain a non-empty feasible set. This previous problem reads:

$$\min_{\Delta \mathbf{u}_f, \boldsymbol{\varepsilon}_{\min}, \boldsymbol{\varepsilon}_{\max}} \|\boldsymbol{\varepsilon}_{\min}\|^2_{\mathbf{W}_{\min}} + \|\boldsymbol{\varepsilon}_{\max}\|^2_{\mathbf{W}_{\max}}$$

$$s.t.$$

$$\Delta \mathbf{u}_{f,\min} \le \Delta \mathbf{u}_f \le \Delta \mathbf{u}_{f,\max}$$

$$\mathbf{u}_{f,\min} \le \boldsymbol{\Pi} \Delta \mathbf{u}_f + \boldsymbol{\Phi} \mathbf{u}_{k-1} \le \mathbf{u}_{f,\max}$$

$$\mathbf{y}_{f,\min} - \boldsymbol{\varepsilon}_{\min} \le \hat{\mathbf{y}}_f(\mathbf{u}_{k-1}, \Delta \mathbf{u}_f) \le \mathbf{y}_{f,\max} + \boldsymbol{\varepsilon}_{\max}$$

where $\mathbf{W}_{\min}$ and $\mathbf{W}_{\max}$ are the inverse of diagonal matrices with the so-called equal concern errors of each limit, and they try to penalize the output constraint violation by penalizing the value of the slack variable. Then, the slack variable solution of this problem, $\boldsymbol{\varepsilon}^*_{\min}$ and $\boldsymbol{\varepsilon}^*_{\max}$, is passed to the main optimization Problem (Eq. (15)), with its last constraint replaced by: $\mathbf{y}_{f,\min} - \boldsymbol{\varepsilon}^*_{\min} \le \hat{\mathbf{y}}_f(\mathbf{u}_{k-1}, \Delta \mathbf{u}_f) \le \mathbf{y}_{f,\max} + \boldsymbol{\varepsilon}^*_{\max}$.

Remark 3. Notice that opposite to the tracking cost term, the economic term explicitly includes the predicted measured variables $\hat{\mathbf{d}}_f$. This is so because some product quality variables are directly related to economic objectives, but they cannot be directly controlled. This way, by a proper selection of $\mathbf{W}y$, $\mathbf{W}u$ and $\mathbf{W}d$, one can choose any combination of controlled variables, $\hat{\mathbf{y}}_f$, and product quality variables, $\hat{\mathbf{d}}_f$, for the economic objective. As it is shown next in the Case Study, a useful possibility is to exclusively consider controlled variable for the tracking objective and product quality (measured variables) for the economic objective.

2.6. Predictions validity indicator

Provided that the performance of any MPC strategy relies on the quality of predictions, validity indicators can be introduced in the LV-MPC problem to evaluate the validity of predictions. Latent variable methods provide indicators of validity of the model referred to the identification dataset. Such indicators can be used to supervise (in parallel form) the model validity and thus avoid using the model for extrapolation. The following combined index, that measures modeled and residual variability [24], can be used as validity indicator:

$$I_{C,k} = \boldsymbol{x}'_{p1,k} \left[\frac{1}{\tau^2_{\alpha,k}} \boldsymbol{P}_{p1,k} \boldsymbol{\Lambda}_k^{-1} \mathbf{P}'_{p1,k} + \frac{1}{\delta^2_{\alpha,k}} \left(\mathbf{I} - \boldsymbol{P}_{p1,k} \boldsymbol{P}'_{p1,k} \right) \right] \boldsymbol{x}_{p1,k}, \tag{17}$$

where $\mathbf{x}_{p1,k} = D^{-1}_{\mathbf{x}_{p1,k}}(\mathbf{x}_{p1,k} - \bar{\mathbf{x}}_{p1,k})$ are the past data, $\boldsymbol{\Lambda}_k$ is the covariance matrix of the latent vector $\mathbf{t}_k = \mathbf{P}'_{p1,k}\mathbf{x}_{p1,k}$, $\tau^2_{\alpha,k}$ and $\delta^2_{\alpha,k}$ are the control limits of each term. When this combined index signals an alert, i.e., $I_{C,k} \ge I_{\alpha,k}$ (with $I_{\alpha,k}$ being a control limit), the economic cost should be eliminated ($\lambda = 0$) from the cost function (see Eq. (15)) so as not to extrapolate excessively from the model. These control limits are estimated using Kernel Density Estimation (KDE) [32,33].

2.7. Model and trajectory updating to drive the process to economic optimal region

In order to reach an economic optimum, the economic LV-MPC controller (Eq. (15)) must be driven mainly by the economic cost, tracking a trajectory different from the original one. However, the MPCA model was initially calibrated around that original set-point trajectory, and so the updating should be done carefully. We propose to get closer to the economic optimal region (batch to batch), by updating the model and trajectory after each new batch has elapsed. To fulfill this aim, the following methodology is proposed:

Table A1
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	−10,000.0 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	10,000
k_2^1	38.9057
k_2^2	17,000
Δt	0.2 min
τ_j	3 min

1 Run the batch process under economic LV-MPC and save new batch data $\mathbf{x}$ (see Eq. (5))
2 Update RBWU dataset $\mathbf{X}$ (Eq. (3)) by including the new batch $\mathbf{x}$
3 Update model (Eq. (4)) by re-calibrating $\mathbf{P}$ with the matrix $\mathbf{X}$ updated
4 Update set-point trajectory as follows:

new set-point = trajectory followed by controlled variables.

5 Compute the cost at the batch end given by $\ell_{total} = (1-\lambda)\ell_{track} + \lambda \ell_{eco}$.

If $\ell_{total}(previousbatch) - \ell_{total}(currentbatch) \ge threshold$,
Re-calculate parameters for monitoring the index $I_{C,k}$ and go to step 1.
else the end.

Low values of λ (e.g., $\lambda = 0.1 - 0.3$) are used to avoid that the economic cost ℓ_{eco} in the cost function ℓ_{total} move the solution to a region where the model is excessively extrapolated. However, the parameter λ may increase with the number of batch runs.

The economic LV-MPC technique is designed to be used when there is not a reliable model that represents the entire workspace of the process. Therefore, an economically optimal trajectory to follow cannot be obtained. In summary, the updating aim is start with an original trajectory (defined arbitrarily) and to arrive (autonomously) at an optimal trajectory after several updates.

3. Case study 1: constrained LV-MPC for tracking of a batch reactor

In this section, a batch reactor is used to illustrate the trajectory tracking capabilities and properties of the proposed algorithm. Aziz et al. [9] presented a nonlinear model of this batch reactor. This process model was originally proposed by Cott and Macchieto [7] 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 productsM_C, M_D are dynamically produced from the quantities of raw materialsM_A, M_B. Values for the model parameters, under nominal conditions, are the same as those reported in Aziz et al. [9] (see Table A1). 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 [9]. 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 Ref.

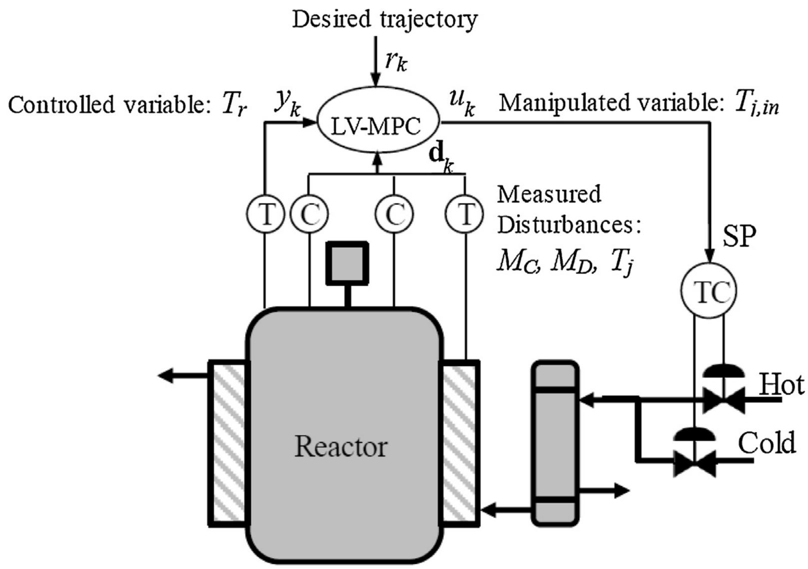

Fig. 1. Schematic of the reactor with its LV-MPC instrumentation given by r_k, y_k, $\mathbf{d}_k$ and u_k. The instrumentation of the PI control (SISO) comprises only r_k, y_k and u_k.

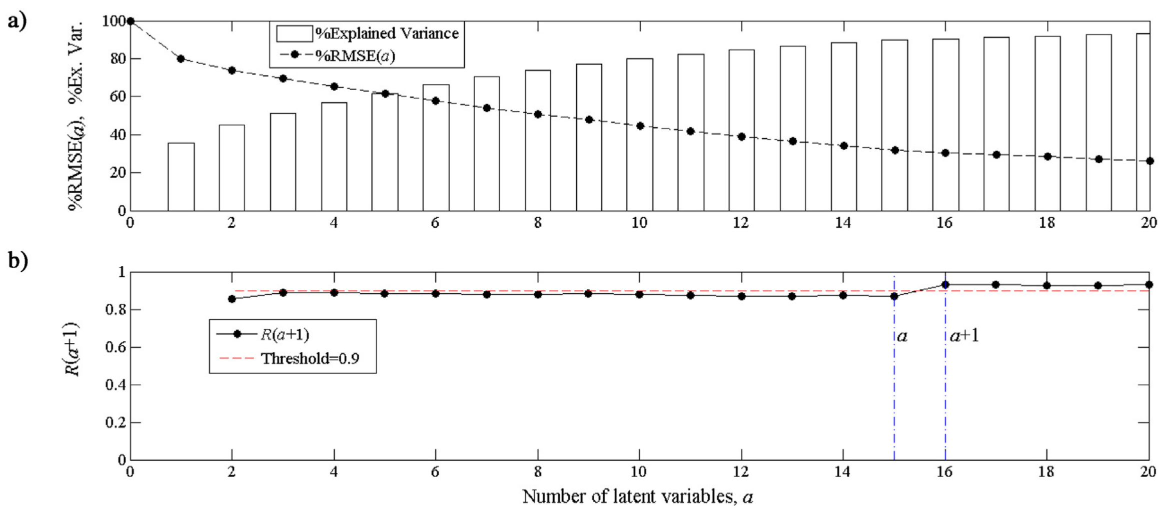

Fig. 2. Selection of the number of latent variables by using the criterion $R_{0.9}$. (a) Pareto of explained variability and percentage of standard error. (b) Successive errors rate together with its threshold for selection of the model order ($A = 15$).

[20]. On-line reactor temperatures, considered to be available every 0.2 min, are corrupted by normally distributed random error with standard deviation $\sigma = 0.15\,^{\circ}$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) [9] 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 ($\pm 12\,^{\circ}$C $\approx \pm 15\%$ 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: $\boldsymbol{x}_k = \left[T_r T_{j,in} M_C M_D T_j\right]'_k = \left[y_k u_k \boldsymbol{d}'_k\right]'$, where $n = 5$. A regularization parameter $L = 5$ was adopted to construct the data matrix $\mathbf{X}$. 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 $MSE(a)$ is the Mean Squared Error using "a" latent variables and $MSE(a) = SSE(a)/(I(L+1)(K-L)n)$. 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 $A = a$. Fig. 2 shows the selection criterion used to set $A = 15$. Fig. 2 shows also the standard error $RMSE(a)$ and the explained variance given by: $R^2_X(a) = tr(\hat{X}'(a)\hat{\mathbf{X}}(a))/tr(\mathbf{X}'\mathbf{X})$, 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 $E_{Tr} = 0.35^{\circ}$C is the allowable mean error and $s = 3.6$ is the move suppression factor.

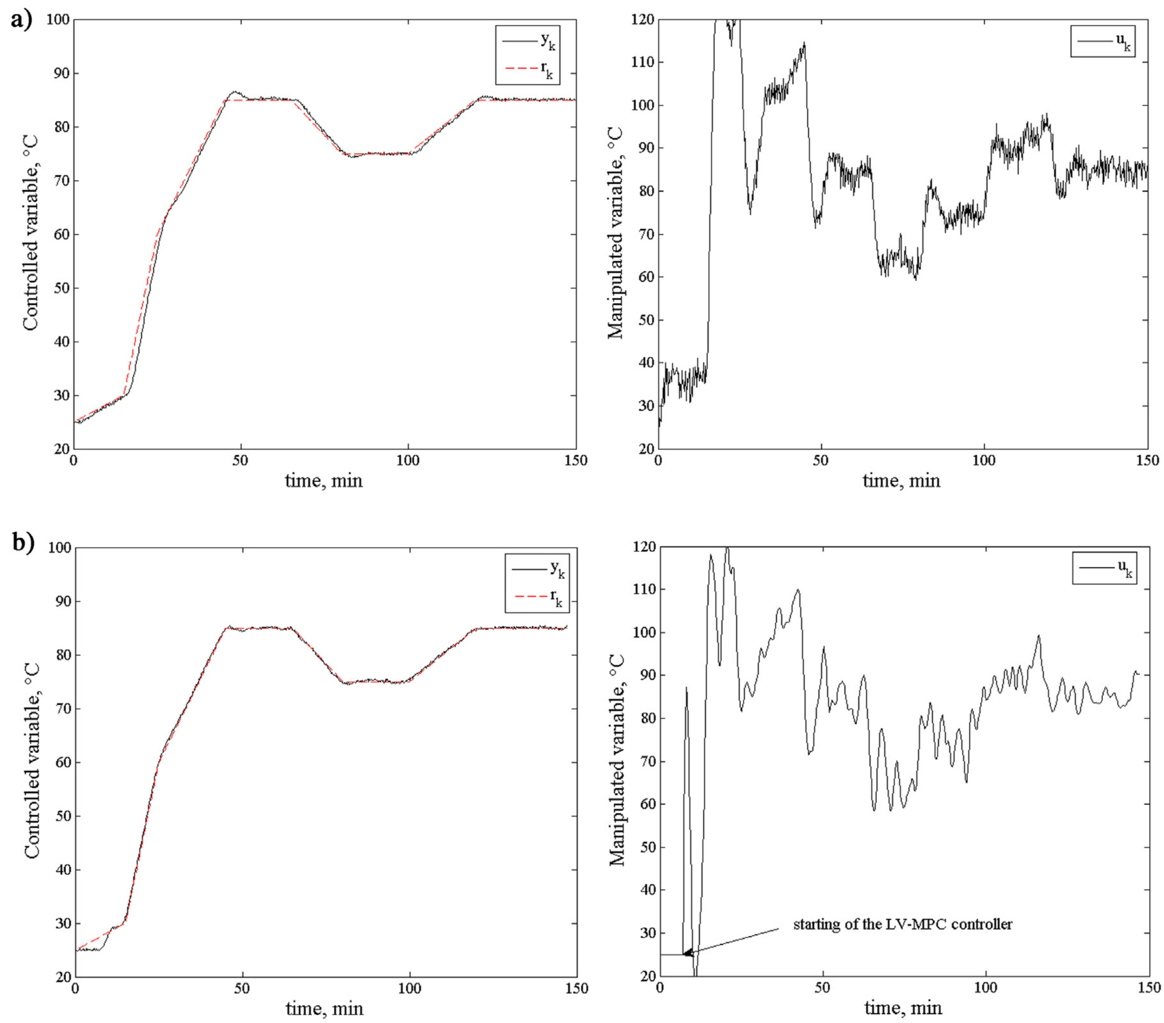

Fig. 3. (a) Implementation of a tightly tuned PI controller for the temperature control problem. (b) Implementation of constrained LV-MPC using a PCA model based on RBWU dataset.

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.

Fig. 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 Ref. [22] (evidenced by its poor performance in the simulations). Fig. 3b shows the trajectory tracking of the constrained LV-MPC during a nominal batch run.

In Refs. [21–23], 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 $\mathbf{P}$) 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. 3b).

The proposed tuning methodology produces a LV-MPC with tighter tracking using a more parsimonious model, which was calibrated using less batch runs than [23]. 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. 3b). 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 Refs. [22,23], the manipulated variable takes negative values (not physically realizable). 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 very large additional random walk disturbance [22,23] 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. Fig. 4 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. However, the proposed control methodology is clearly able to reject the non-stationary disturbances. The input moves in Fig. 4

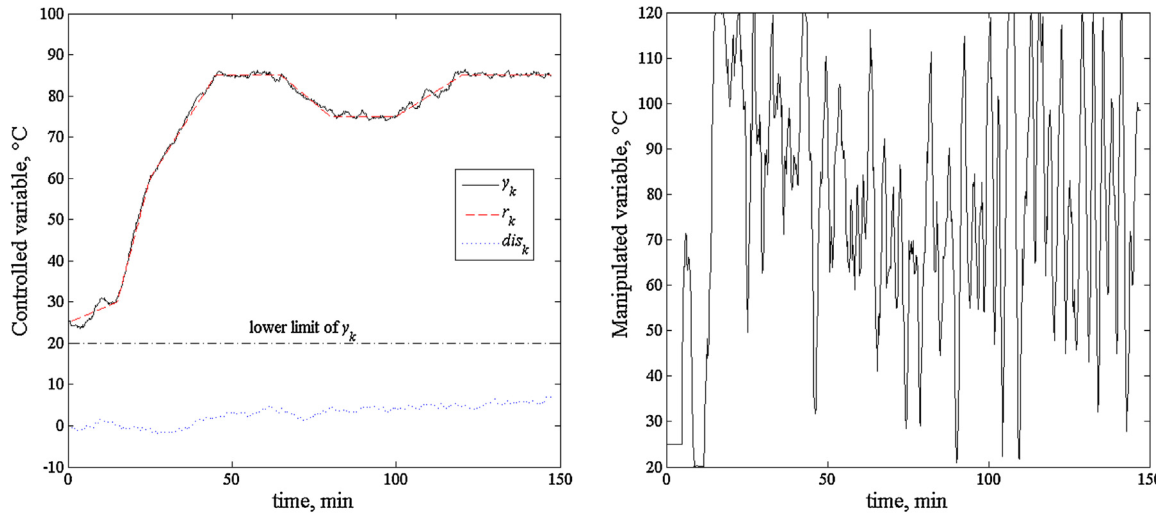

Fig. 4. Performance of LV-MPC for both tracking the set-point trajectory and rejecting a random walk disturbance occurring on top of the output (T_r).

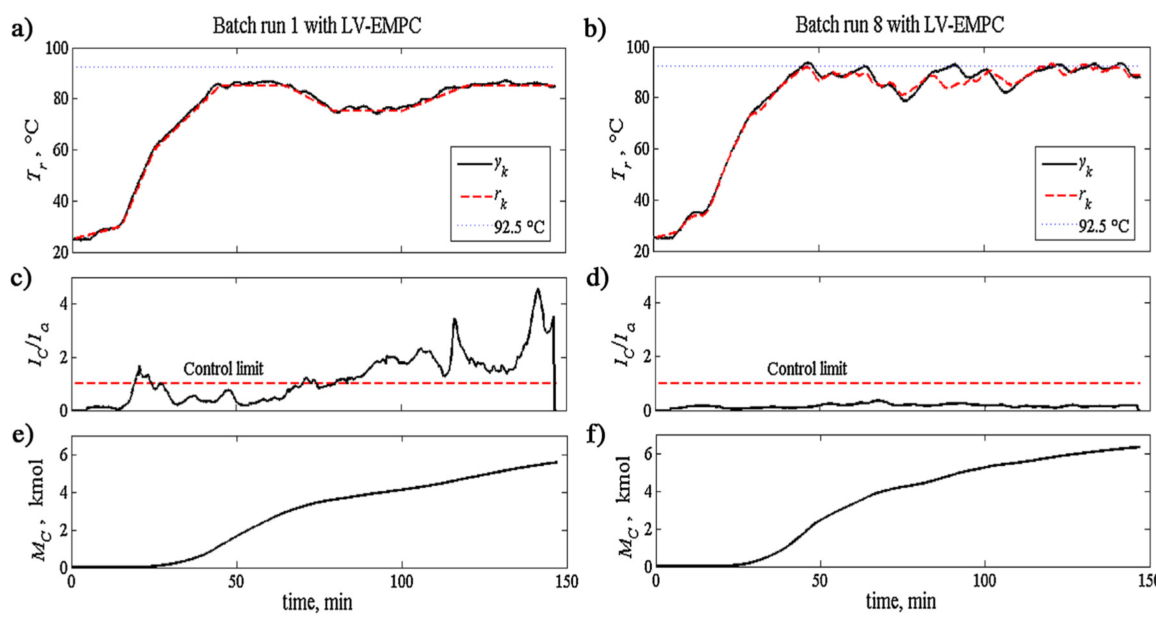

Fig. 5. Batch runs using LV-EMPC. Time evolutions of reactor temperature (and its set-point), model validity indicator and desired product (maximized variable).

are more aggressive than in Fig. 3, 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 very 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. Case study 2: economic LV-MPC with trajectory and model updating

The case study is a batch reactor in which the following exothermic reactions take place

$$A + B \rightarrow C, A + C \rightarrow D,$$

where A, B are the raw materials, C is the desired product and D is the waste product. Then, the following economic cost function, that includes only the predicted measured variables $\hat{\mathbf{d}}_f$, can be proposed:

$$\ell_{eco} = \|\mathbf{W}_{eco}(\boldsymbol{d}_{f,opt} - \hat{\mathbf{d}}_f)\|^2,$$

where $\mathbf{W}_{eco} = (1/E_{MC}PH^{0.5})\mathbf{I}_{PH}$ are the weights used with $E_{MC} = 0.01 kmol$ being the allowable mean error, $\hat{\mathbf{d}}'_f = [\hat{M}_{C,k} \cdots \hat{M}_{C,k+PH-1}]$ and $\mathbf{d}'_{f,opt} = M_{C,opt}\,[1 \ldots 1] \in \Re^{1\times PH}$ with $M_{C,opt} = 10 kmol$ being the desired end product. Thus, the total cost function for this case study is given by:

$$\ell_{total} = (1 - \lambda)\ell_{track}(\hat{\boldsymbol{y}}_f, \boldsymbol{r}_f, \Delta\boldsymbol{u}_f) + \lambda\ell_{eco}(\hat{\mathbf{d}}_f),$$

where the tracking cost function ℓ_{track} has the same parameters as the previous case study and $\lambda = 0.25$. The economic LV-MPC with trajectory and model updating is applied using the followings constraints: $20 \leq T_r \leq 100$, $20 \leq T_{j,in} \leq 120$, $-15 \leq \Delta T_{j,in} \leq 15$. Fig. 5 shows the time evolution of the reactor temperature (and its set-point), the model validity indicator, and the desired product, by using economic LV-MPC during batch runs 1 and 8. For this process, it is known that the higher the temperature, the more desired product C is obtained. The optimum temperature of the reactor should be between 90 and 100 °C [7]. The evolution of the reactor temperature coincides with this behavior; the economic cost tries to raise the reactor temperature while the tracking cost tries to keep the temperature close to the set-point (see Fig. 5a). The model validity indicator claims that the model became more unreliable at the end of the batch (see Fig. 5c), just when the desired product is further away from the historical behavior (see Fig. 5e). The model updating was carried out during 7 new batch runs in order to drive the process to an economically better region. Fig. 5 shows also the time evolution of the controlled variable (see Fig. 5b), the model validity indicator (Fig. 5d), and the desired product (Fig. 5f), during the batch run 8. Fig. 5b shows that the set-point trajectory has been modified compared to the original one (see Fig. 5a) and that the model became significantly more reliable, with a significantly lower indicator (see Fig. 5d), in comparison with the batch run 1 (see Fig. 5c). This is so because the model was moved to a new economically optimum region. In Ref. [34] it was determined that the optimum temperature profile should rise quickly and then remains constant at a value close to 92.5 °C. The updated temperature trajectory tends to this behavior (see Fig. 5b). The end product obtained by using LV-MPC is M_C = 5.5 kmol, which corresponds to an end economic cost of $\ell_{eco} = 202,500$. The end product obtained by using economic LV-MPC after the batch run 8 is M_C = 6.43 kmol (Fig. 5f), which correspond to an end economic cost of $\ell_{eco} = 127,449$. Therefore, a significant increase in the desired product is verified (17%), and an improvement (reduction) of the economic cost of 37%. In this simulation, the economic cost is not suppressed (by $\lambda = 0$) when the indicator is on alert (i.e., when $I_{C,k}/I_{\alpha,k} \geq 1$). However, the updating strategy was effective.

It is important to highlight that the updating strategy used is a first approximation (the simplest). Other strategies could be more effective. For instance, it is possible to update the RBWU dataset $\mathbf{X}$ by including the new batch $\mathbf{x}$ and excluding the most distant batch $\mathbf{x}_i$, which could be identified by: $\boldsymbol{x}_i = arg\max_{i=1\ldots I}(\mathbf{x}_i - \bar{\mathbf{x}})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \bar{\mathbf{x}})$, where $\bar{\mathbf{x}}$ is the mean batch and $\mathbf{\Sigma}$ is the covariance matrix of the BWU dataset $\mathbf{X}_B$ (both population parameters would be estimated including the new batch). In general, other structures of optimization-control-update could also be used, that could be even more effective. However, only a first approximation was simulated in order to show the main elements involved in the proposed controller and to illustrate the good performance that can be obtained.

5. Conclusions

This paper presented a constrained LV-MPC technique for trajectory tracking and economic optimization in batch process. The economic LV-MPC technique includes economic objectives in the cost formulation and it is based on trajectory and model updating (batch to batch) in order to autonomously find an economic optimum. This technique is the first approach of economic LV-MPC that has been presented and we believe it has a promising future.

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 very 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 have QP solvers, and hence, the offset-free LV-MPC formulation could be easily implemented. Two case studies have 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 develop of a reliable method to compute the number of shifts in matrix $\mathbf{X}$, 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; (iii) the application of economic LV-MPC to bioreactors; (iv) the use of different structures of optimization-control-update, including the updating strategy.

Acknowledgements

The authors are grateful for the financial support received from CAPES, CNPq, and CONICET.

Appendix A. Process model for the case study

The model equations for the batch reactor are as follows:

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

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

$$R_1 = k_1 M_A M_B \tag{A.3}$$

$$R_2 = k_2 M_A M_C \tag{A.4}$$

$$Q_r = -\Delta H_1 R_1 - \Delta H_2 R_2 \tag{A.5}$$

$$M_r = M_A + M_B + M_C + M_D \tag{A.6}$$

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

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

$$\frac{dM_A}{dt} = -R_1 - R_2 \tag{A.9}$$

$$\frac{dM_B}{dt} = -R_1 \tag{A.10}$$

$$\frac{dM_C}{dt} = +R_1 - R_2 \tag{A.11}$$

$$\frac{dM_D}{dt} = +R_2 \tag{A.12}$$

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

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

The parameters of the above model are given in Table A1.

References

[1] G.M. Troup, C. Georgakis, Process systems engineering tools in the pharmaceutical industry, Comput. Chem. Eng. 51 (2013) 157–171.
[2] T. Clarke-Pringle, J.F. MacGregor, Nonlinear adaptive temperature control of multi-product, semi-batch polymerization reactors, Comput. Chem. Eng. 21 (1997) 1395–1409.
[3] H.C. Lontra, Modelagem e simulaça˜o de um reator batelada de polimerizaça˜o de estireno em suspensa˜o, in: Anais Do Congresso Brasileiro De Polımeros, Sao˜ Paulo, Brazil, 1991, pp. 619–623.
[4] A. Altintena, F. Ketevanlioglua, S. Erdog̃ana, H. Hapog̃lub, M. Alpbaz, Self-tuning PID control of jacketed batch polystyrene reactor using genetic algorithm, Chem. Eng. J. 38 (2008) 490–497.
[5] C. Kravaris, R.A. Wright, J.F. Carrier, Nonlinear controllers for trajectory tracking in batch processes, Comput. Chem. Eng. 13 (1989) 73–82.
[6] C. Kravaris, M. Saroush, Synthesis of multivariate nonlinear controllers by input/output linearization, AIChE J. 36 (1990) 249–264.
[7] B.J. Cott, S. Macchietto, Temperature control of exothermic batch reactors using generic model control, Ind. Eng. Chem. Res. 28 (1989) 1177–1184.
[8] Z.L. Wang, F. Pla, J.P. Corriou, Nonlinear adaptive control of batch styrene polymerization, Chem. Eng. Sci. 50 (1995) 2081–2091.
[9] N. Aziz, M.A. Hussain, I.M. Mujtaba, Performance of different types of controllers in tracking optimal temperature profiles in batch reactors, Comput. Chem. Eng. 24 (2000) 1069–1075.

[10] C.E. García, Quadratic dynamic matrix control of nonlinear processes. an application to a batch reaction process, in: AIChE Annual Meeting, San Francisco, CA, 1984.
[11] G. Gattu, E. Zafiriou, Nonlinear dynamic matrix control with state estimation, Ind. Eng. Chem. Res. 31 (1992) 1096–1104.
[12] T. Peterson, E. Hernandez, Y. Arkun, J.F. Schork, A nonlinear DMC algorithm and its application to a semi-batch polymerization reactor, Chem. Eng. Sci. 47 (1992) 737–753.
[13] E.F. Camacho, C. Bordons, Model Predictive Control, 2nd ed., Springer-Verlag, 2004.
[14] J.B. Rawlings, D.Q. Mayne, Model Predictive Control: Theory and Design, 1st ed., Nob-Hill Publishing, 2009.
[15] A. Ferramosca, A.H. González, D. Limon, G.A. Bustos, J.L. Godoy, J.L. Marchetti, On economic optimality of model predictive control, IEEE Lat. Am. Trans. 12 (2014) 1234–1241.
[16] A.C. Zanin, M. Tvrzska de Gouva, D. Odloak, Integrating real time optimization into the model predictive controller of the FCC system, Control Eng. Pract. 10 (2002) 819–831.
[17] P. Nomikos, J.F. MacGregor, Monitoring batch processes using multiway principal component analysis, AIChE J. 40 (1994) 1361–1375.
[18] P.P.C. Nelson, P.A. Taylor, J.F. MacGregor, Missing data methods in PCA and PLS: score calculations with incomplete observations, Chemom. Intell. Lab. Syst. 35 (1996) 45–65.
[19] J. Flores-Cerrillo, J.F. MacGregor, Latent Variable MPC for trajectory tracking in batch processes, J. Process Control 15 (2005) 651–663.
[20] H. Yu, J. Flores-Cerrillo, Latent variable model predictive control for trajectory tracking in batch processes: internal model control interpretation and design methodology, Ind. Eng. Chem. Res. 52 (2013) 12437–12450.
[21] J.F. MacGregor, M.J. Bruwer, M. Golshan, System and methodology for the model predictive control of batch processes using latent variable dynamic models, U.S. Patent 61/052,9922008.
[22] M. Golshan, J.F. MacGregor, M.J. Bruwer, P. Mhaskar, Latent variable model predictive control (LV-MPC) for trajectory tracking in batch processes, J. Process Control 20 (2010) 538–550.
[23] M. Golshan, J.F. MacGregor, P. Mhaskar, Latent variable model predictive control for trajectory tracking in batch processes: alternative modeling approaches, J. Process Control 21 (2011) 1345–1358.
[24] J.L. Godoy, J.R. Vega, J.L. Marchetti, Relationships between PCA and PLS-regression, Chemom. Intell. Lab. Syst. 130 (2014) 182–191.
[25] A.H. González, E.J. Adam, J.L. Marchetti, Conditions for offset elimination in state space receding horizon controllers: a tutorial analysis, Chem. Eng. Process. 47 (2008) 2184–2194.
[26] S.J. Qin, T.A. Badgwell, A survey of industrial model predictive control technology, Control Eng. Pract. 11 (2003) 733–764.
[27] C. Duchesne, J.F. MacGregor, Jackknife and bootstrap methods in the identification of dynamic models, J. Process Control 11 (2001) 553–564.
[28] M. Diehl, R. Amrit, J.B. Rawlings, A Lyapunov function for economic optimizing model predictive control, IEEE Trans. Autom. Control 56 (2011) 703–707.
[29] D. Angeli, R. Amrit, J.B. Rawlings, On average performance and stability of economic model predictive control, IEEE Trans. Autom. Control 57 (2012) 1615–1626.
[30] A. Ferramosca, D. Limon, A.H. González, I. Alvarado, E.F. Camacho, Robust MPC for tracking zone regions based on nominal predictions, J. Process Control 22 (2012) 1966–1974.
[31] T. Alamo, A. Ferramosca, A.H. González, D. Limon, D. Odloak, A gradient-based strategy for the one-layer RTO + MPC controller, J. Process Control 24 (2014) 435–447.
[32] C. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.
[33] E. Martin, A. Morris, Non-parametric confidence bounds for process performance monitoring charts, J. Process Control 6 (1996) 349–358.
[34] I.M. Mujtaba, N. Aziz, M.A. Hussain, Neural network based modeling and control in batch reactor, Chem. Eng. Res. Des. 84 (2006) 635–644.