Journal of Process Control 43 (2016) 1–9

Contents lists available at ScienceDirect

Journal of Process Control

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

Just-in-Time-Learning based Extended Prediction Self-Adaptive Control for batch processes☆

Qing-Lin Su[a], Martin Wijaya Hermanto[b], Richard D. Braatz[c], Min-Sen Chiu[a,*]

[a] Department of Chemical and Biomolecular Engineering, National University of Singapore, 117576, Singapore
[b] Institute of Chemical and Engineering Sciences, 1 Pesek Road, Jurong Island 627833, Singapore
[c] Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

ARTICLE INFO

Article history:
Received 14 May 2015
Received in revised form 29 February 2016
Accepted 13 April 2016
Available online 10 May 2016

Keywords:
Just-in-Time Learning
Local models
Nonlinear model predictive control
Batch processes
EPSAC

ABSTRACT

This article presents a new Extended Prediction Self-Adaptive Control (EPSAC) algorithm based on the Just-in-Time Learning (JITL) method. In the proposed JITL-based EPSAC design, linearization of the process model is achieved by a set of local state-space models, each of which can be independently and simultaneously identified by the JITL method along the base trajectory. For the end-product quality control for a simulated semi-batch pH-shift reactive crystallization process where shrinking prediction and control horizons are essential, the proposed EPSAC algorithm not only simplifies the control weight tuning but also provides better and more robust closed-loop control performance than its previous counterpart.

1. Introduction

With the stringent specifications on product quality and higher competition in the process industry, the development of nonlinear model predictive control (NMPC) is of interest to both academic and industrial sectors. The major advantage of NMPC lies in its capability of handling nonlinearities and time-varying characteristics inherent in the process dynamics in addition to addressing constraints and bounds imposed on both state and manipulated variables by performing the real-time dynamic optimization [1,2]. Various NMPC design methods have been developed using different techniques to deal with process nonlinearity, including successive linearization [3], neural networks [4], robust control [5,6], multiple models [7–10] and hybrid models [11].

Among various NMPC design methods reported in the literature, Extended Prediction Self-Adaptive Control (EPSAC) and its variants adopt the approach of iterative optimization based on a predefined input trajectory. [12–15]. In the EPSAC design framework, the model prediction consists of a base and an optimized term. The base term is computed from a nominal process model using the current values of input variables obtained from the predefined base input trajectory and the corresponding output variables, while the optimized term is computed from a finite step or impulse response model obtained along this trajectory. In this manner, the nonlinear process model can be directly used in the form of an input–output algorithm without any linearization [13,16,17].

A potential drawback of previous EPSAC methods is the incorporation of a convolution model in the formulation of the control algorithms. Since model parameters are obtained by introducing a step change to the current input value specified by the base input trajectory, the predicted outputs at sampling instants further away from the current sampling instant become less accurate due to process nonlinearity, leading to inevitable modeling error that degrades the achievable closed-loop performance. To alleviate the abovementioned limitation, the aim of this paper is to develop a new EPSAC algorithm using a set of local state-space models identified by the Just-in-Time Learning (JITL) method [18,19] along the base trajectory. To evaluate the performance of proposed design for batch process control, simulation results of implementing the JITL-based EPSAC to a semi-batch pH-shift reactive crystallization process are presented and discussed in detail.

2. Conventional EPSAC algorithm

NMPC refers to predictive control schemes based on nonlinear models and/or a nonlinear cost function and nonlinear constraints,

☆ The proposed EPSAC algorithm was partially demonstrated at the 2012 AIChE Annual Meeting in Pittsburgh, U.S.

* Corresponding author. Fax: +65 6779 1936.
E-mail address: checms@nus.edu.sg (M.-S. Chiu).

http://dx.doi.org/10.1016/j.jprocont.2016.04.009
0959-1524/

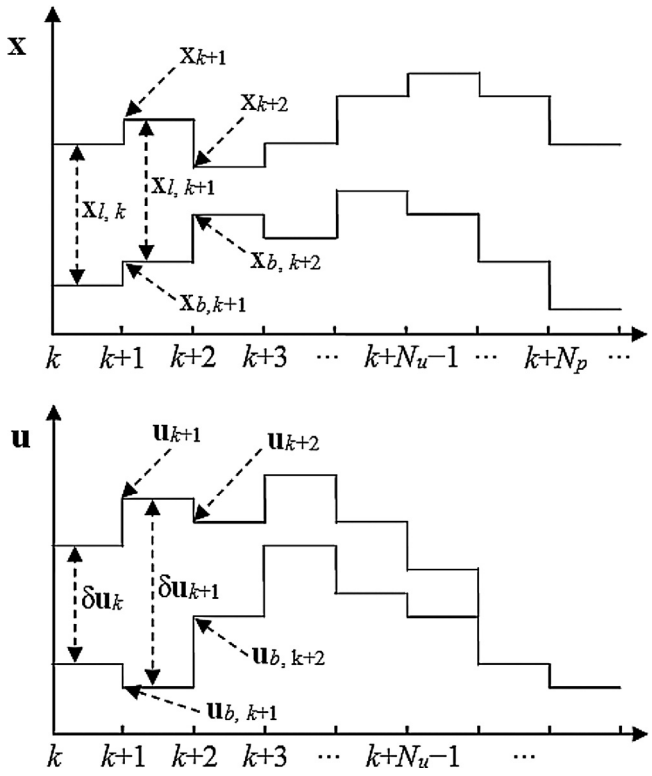

Fig. 1. The variables decomposition in the EPSAC method.

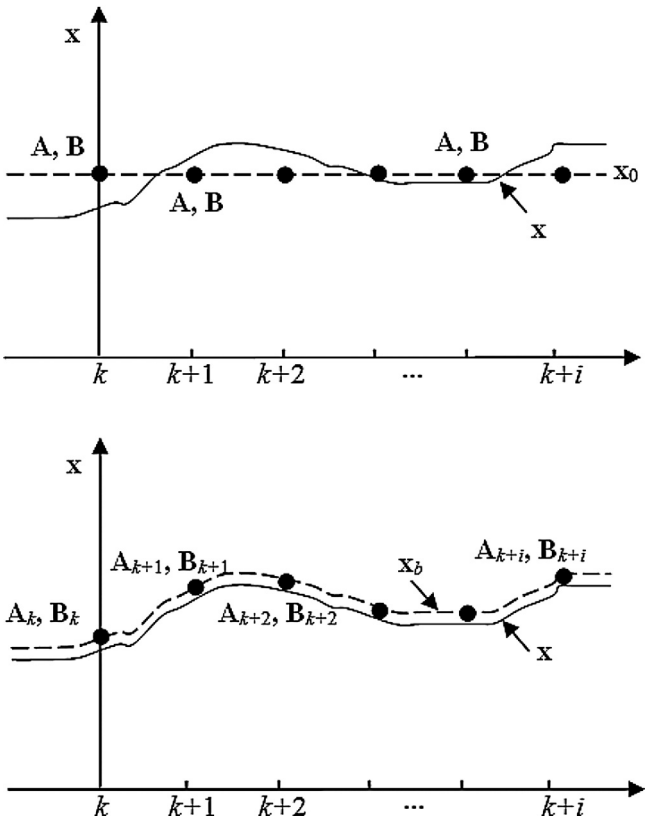

Fig. 2. Modeling of nonlinear processes using one local state-space model (top) and multiple state-space models (bottom).

whose optimal problem to be solved online at every sampling instant is [15]

$$\min_{\boldsymbol{u}_k,\ldots,\boldsymbol{u}_{k+Nu-1}} \quad J(\boldsymbol{x},\boldsymbol{u}) \tag{1}$$

subject to

$$x_{k+i} = f(\boldsymbol{x}_{k+i-1}, u_{k+i-1}) + \boldsymbol{w}_{k+i}, \quad i = 1,\ldots,N_u,\ldots,N_p \tag{2}$$

$$\boldsymbol{d}_{k+i} = \boldsymbol{d}_{k+i-1} + \boldsymbol{\xi}_{k+i} \tag{3}$$

$$\boldsymbol{y}_{k+i} = g(\boldsymbol{x}_{k+i}, \boldsymbol{u}_{k+i}) + \boldsymbol{d}_{k+i} + \boldsymbol{v}_{k+i} \tag{4}$$

$$h(\boldsymbol{x}_{k+i}, \boldsymbol{u}_{k+i}) \le \boldsymbol{0} \tag{5}$$

where J is the objective function, N_u is the control horizon, N_p is the prediction horizon, $\mathbf{x}_k$, $\mathbf{u}_k$, $\mathbf{y}_k$, and $\mathbf{d}_k$ are the vectors of n_x system state variables, n_u inputs, n_y measured variables, and n_y unmeasured disturbances at the kth sampling instant, respectively, and $\mathbf{w}_k$, $\boldsymbol{\xi}_k$, and $\mathbf{v}_k$ are the vectors of noise on the state variables, unmeasured disturbances, and the measured variables, respectively. The system dynamics are described by the vector function f, the measurement equations by the vector function g, and the linear and nonlinear constraints for the system are described by the vector function h.

In the EPSAC strategy, future response can be expressed as the cumulative result of two effects: (1) a base response that accounts for the effect of past control, a base future control scenario, and the effect of future disturbances; and (2) an optimizing response that accounts for the effect of the optimizing future control actions [14], as schematically illustrated in Fig. 1. The future sequences of the input variables $\mathbf{u}_{k+i}$ are considered as the sum of the base input $\mathbf{u}_{b,\,k+i}$ and future incremental control actions $\delta\mathbf{u}_{k+i}$:

$$u_{k+i} = \boldsymbol{u}_{b,\,k+i} + \delta\boldsymbol{u}_{k+i}, \quad i = 0, 1,\ldots, N_u - 1 \tag{6}$$

where $\delta\mathbf{u}_{k+i} = \mathbf{0}$ for $i \ge N_u$. Then the future trajectories of process variables can also be considered as the cumulative result of these two effects:

$$\mathbf{x}_{k+i} = \boldsymbol{x}_{b,\,k+i} + \boldsymbol{x}_{l,\,k+i}, \quad i = 1, 2,\ldots, N_p \tag{7}$$

where $\mathbf{x}_{b,\,k+i}$ is calculated using the nominal model of Eq. (2) and the predetermined sequence of $\mathbf{u}_{b,k+i}$. On the other hand, $\mathbf{x}_{l,\,k+i}$ is obtained by implementing impulse inputs $\{\delta\mathbf{u}_k,\ \delta\mathbf{u}_{k+1},\ldots,\ \delta\mathbf{u}_{k+i-1}\}$. A similar decomposition into the sum of two parts is also applied to the nonlinear constraints of Eq. (5) to arrive at a quadratic program (QP) problem, where the soft constraint approach is used to provide a numerical convergence of QP optimizer [15,20].

The key idea of EPSAC is to predict nonlinear process variables by iterative optimization with respect to future trajectories so that they converge to the same nonlinear optimal solution [14,15,21]. The conventional EPSAC algorithms use convolution models, such as finite step or impulse models, to predict process outputs around the future trajectories ($\mathbf{x}_b$ and $\mathbf{u}_b$, see Fig. 1) in order to calculate the optimized term $\mathbf{x}_{l,\,k+i}$ in Eq. (7) from $\delta\mathbf{u}_{k+i}$ in Eq. (6).

The superposition principle underlying the predictive schemes given in Eqs. (6) and (7), which is employed by previous EPSAC designs, becomes unreliable and hence inaccurate for long prediction horizons in the presence of nonlinear process dynamics and time-varying process characteristics. To partially address this shortcoming brought about by using the convolution models in the EPSAC design, complicated weights in the objective function have been designed and fine-tuned to improve the end product quality control of a batch crystallization process [15].

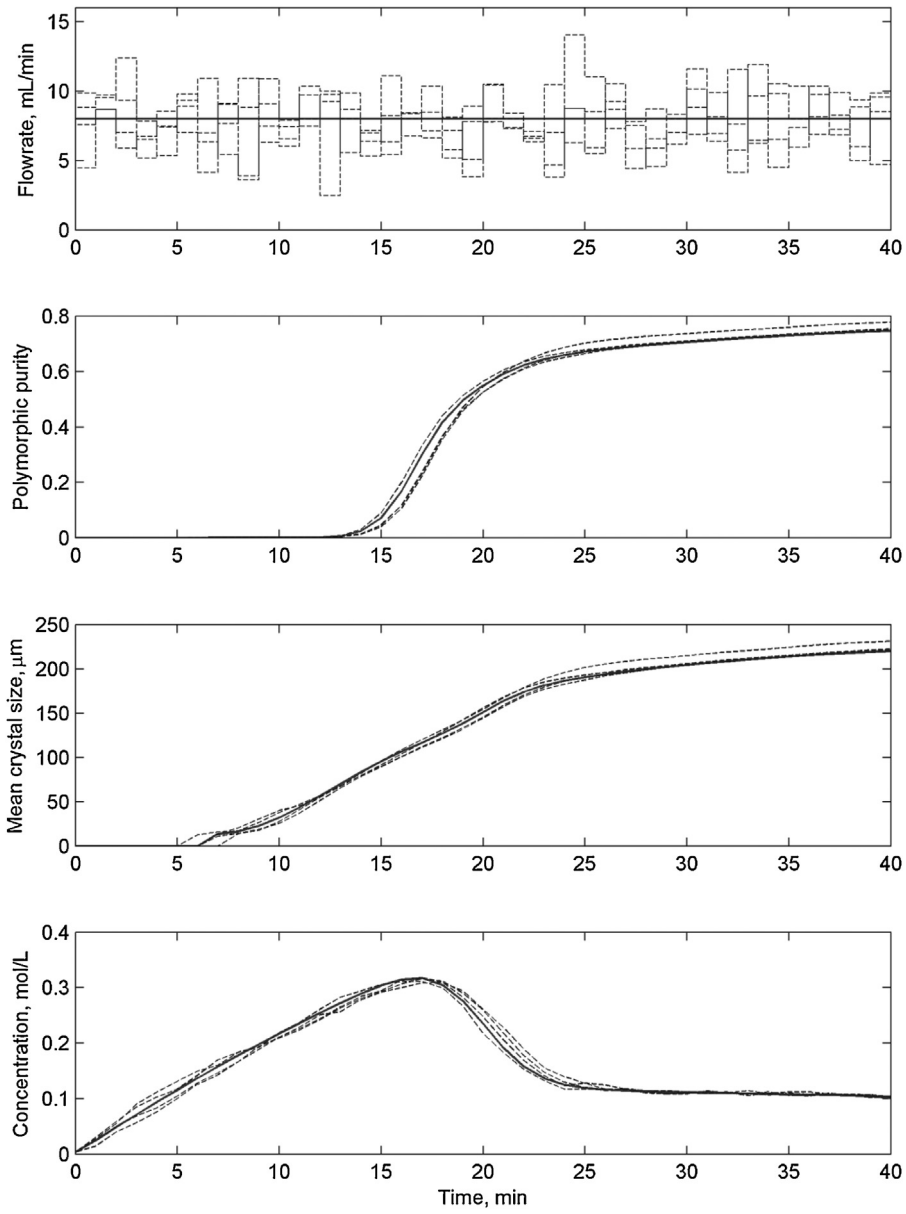

Fig. 3. Illustration of the process data (solid line: nominal data; dashed lines: four batches process data) used for the reference database by the JITL method.

3. The JITL-based EPSAC design

3.1. JITL local state-space model

Linear or nonlinear processes operated within a narrow range of a steady-state operating point, i.e., $\{\mathbf{x}_0,\ \mathbf{u}_0\}$, can be described by the state-space model

$$\tilde{\mathbf{x}}_{k+i+1} = A\tilde{\boldsymbol{x}}_{k+i} + \boldsymbol{B}\tilde{\boldsymbol{u}}_{k+i} \tag{8}$$

where $\tilde{\mathbf{x}}_{k+i}$ is the deviation between system state vector $\mathbf{x}$ and the chosen operating point $\mathbf{x}_0$, viz., $\tilde{\mathbf{x}}_{k+i} = \mathbf{x}_{k+i} - \mathbf{x}_0$, at kth sampling instant, the deviation variable for process input vector is also defined accordingly, $\tilde{\mathbf{u}}_{k+i} = \mathbf{u}_{k+i} - \mathbf{u}_0$, $\mathbf{A}$ is a $n_x \times n_x$ matrix, and $\mathbf{B}$ is a $n_x \times n_u$ matrix.

Likewise, considering the reference trajectory of $\{\mathbf{x}_{b,\,k+i},\ \mathbf{u}_{b,\,k+i}\}$ in Fig. 1, a series of local state-space models can also be obtained at each sampling instant in the prediction horizon:

$$\boldsymbol{x}_{l,\ k+i+1} = \boldsymbol{A}_{k+i}\boldsymbol{x}_{l,\ k+i} + \boldsymbol{B}_{k+i}\delta\boldsymbol{u}_{k+i} \tag{9}$$

The implementation of using Eqs. (8) and (9) in modeling is illustrated in Fig. 2, where Eq. (8) employs a single linear model obtained at point $\mathbf{x}_0$ to approximate process nonlinearity, while Eq. (9) employs a set of linear local models constructed for each sampling instant in the reference trajectory $\mathbf{x}_b$ for the same purpose. It is clear from Fig. 2 that the latter gives a more accurate description of the base trajectory. Thus, Eq. (9) is inherently more advantageous than Eq. (8) for nonlinear trajectory modeling and therefore is more suitable for the ESPAC design as it can lessen the aforementioned drawbacks resulting from the use of convolution models in majority of ESPAC designs [22,23].

In this work, the JITL method [18,19] is implemented to obtain a set of models given in Eq. (9). The JITL method has received increasing research interest and has been applied to soft sensing [24], process monitoring [25,26], and controller design [27,28]. The JITL method is a data-based methodology to approximate a nonlinear system by a local model valid in the relevant operating regimes. When a query data comes, JIYL prediction generally involves three steps, viz., relevant data selection from reference database accord-

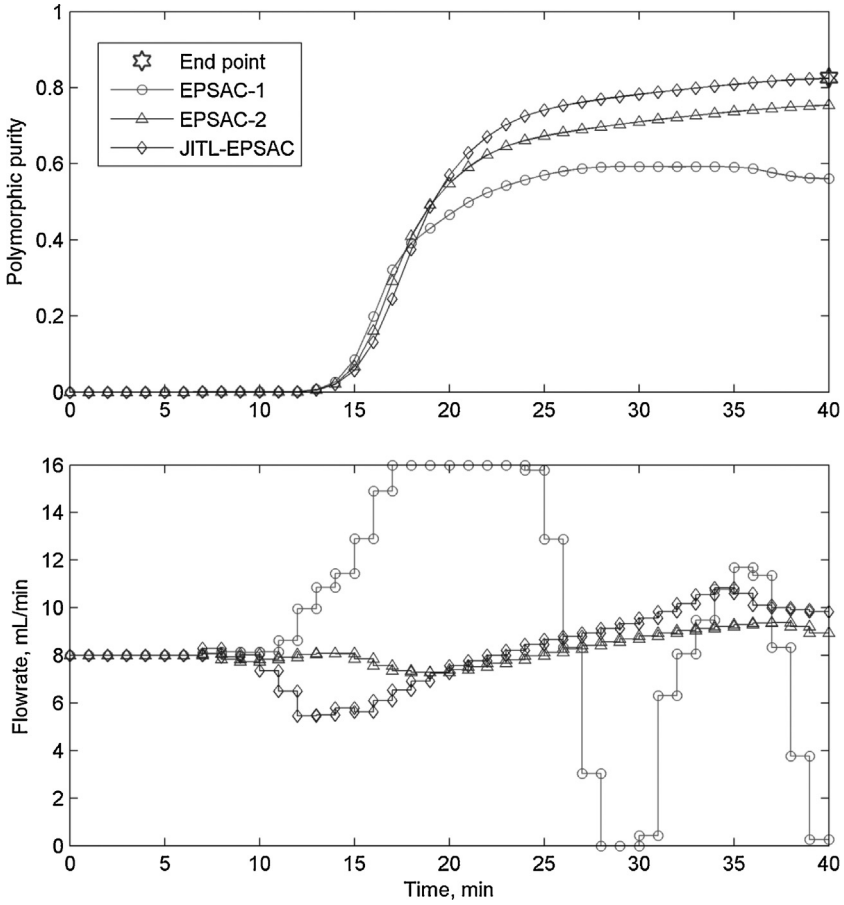

Fig. 4. Performance comparison between the EPSAC and JITL-EPSAC.

ing to some similarity criterion, local model construction based on the relevant data, and model prediction using the local model.

Without the loss of generality, consider the 2×2 state-space model:

$$\begin{bmatrix} x^1_{l,k+1} \\ x^2_{l,k+1} \end{bmatrix} = \begin{bmatrix} a^{1,1}_k & a^{1,2}_k \\ a^{2,1}_k & a^{2,2}_k \end{bmatrix} \begin{bmatrix} x^1_{l,k} \\ x^2_{l,k} \end{bmatrix} + \begin{bmatrix} b^{1,1}_k & b^{1,2}_k \\ b^{2,1}_k & b^{2,2}_k \end{bmatrix} \begin{bmatrix} \delta u^1_k \\ \delta u^2_k \end{bmatrix} \tag{10}$$

where $x^i_{l,k}$ is the ith state variable in the optimized term $\mathbf{x}_{l,k}$ and the remaining symbols have obvious definitions. Then a reference database $\{\mathbf{Y}, \boldsymbol{\Psi}\}$ designed for the first state variable x^1 can be constructed from process data as

$$Y = [x^1_2,\ x^1_3,\ \cdots,\ x^1_n]^T \tag{11}$$

$$\boldsymbol{\Psi} = \begin{bmatrix} x^1_1 & x^2_1 & u^1_1 & u^2_1 \\ x^1_2 & x^2_2 & u^1_2 & u^2_2 \\ \vdots & \vdots & \vdots & \vdots \\ x^1_{n-1} & x^2_{n-1} & u^1_{n-1} & u^2_{n-1} \end{bmatrix} \tag{12}$$

where each corresponding row in $\mathbf{Y}$ and $\boldsymbol{\Psi}$ together represent a sample data and n is the number of sample points collected in the reference database.

During the implementation of the proposed JITL-based EPSAC design, for a query data $\mathbf{q}_k = [x^1_{b,k},\ x^2_{b,k},\ u^1_{b,k},\ u^2_{b,k}]$ obtained at the kth sampling instant from the reference trajectories of $\mathbf{x}_b$and $\mathbf{u}_b$, a relevant dataset $\{\mathbf{Y}_l, \boldsymbol{\Psi}_l\}$ consisting of l samples is selected from $\{\mathbf{Y}, \boldsymbol{\Psi}\}$ according to a chosen similarity criterion. Next, by subtracting $\mathbf{x}_{b,\,k+1}$ and $\mathbf{q}_k$ from each row of $\mathbf{Y}_l$ and $\boldsymbol{\Psi}_l$, respectively, which are denoted as $\tilde{\mathbf{Y}}_l$ and $\tilde{\boldsymbol{\Psi}}_l$, all the sample data relevant to the kth operating point are now in the same form as the optimized term, viz., deviation variables. Therefore, coefficients for the first state variable can be obtained using various regression methods, for example, simply using least squares:

$$[a^{1,1}_k,\ a^{1,2}_k,\ b^{1,1}_k,\ b^{1,2}_k]^T = (\tilde{\boldsymbol{\Psi}}^{\mathrm{T}}_1 \tilde{\Psi}_l)^{-1} \tilde{\boldsymbol{\Psi}}^{\mathrm{T}}_1 \tilde{\mathbf{Y}}_l \tag{13}$$

The aforementioned procedure can be similarly applied to the second state variable x^2 in Eq. (10) to obtain the corresponding coefficients and hence the matrices $\mathbf{A}_k$ and $\mathbf{B}_k$. The detailed information on the JITL methodology can be referred to the latest JITL algorithm by Su et al. [19], which is termed as the JITL with reference to the query point.

3.2. The proposed EPSAC algorithm

After obtaining the local state-space models in the base trajectories $\mathbf{x}_b$ and $\mathbf{u}_b$, the sequence of optimized term $\mathbf{x}_{l,\,k+i}$ in the prediction horizon N_p and control horizon N_u can be calculated as

$$\mathbf{x}_{l,\,k+1} = \mathbf{A}_k \mathbf{x}_{l,\,k} + \mathbf{B}_k \delta \mathbf{u}_k$$

$$\mathbf{x}_{l,k+2} = \mathbf{A}_{k+1}\mathbf{A}_k \mathbf{x}_{l,k} + \mathbf{A}_{k+1}\mathbf{B}_k \delta \mathbf{u}_k + \mathbf{B}_{k+1} \delta \mathbf{u}_{k+1}$$

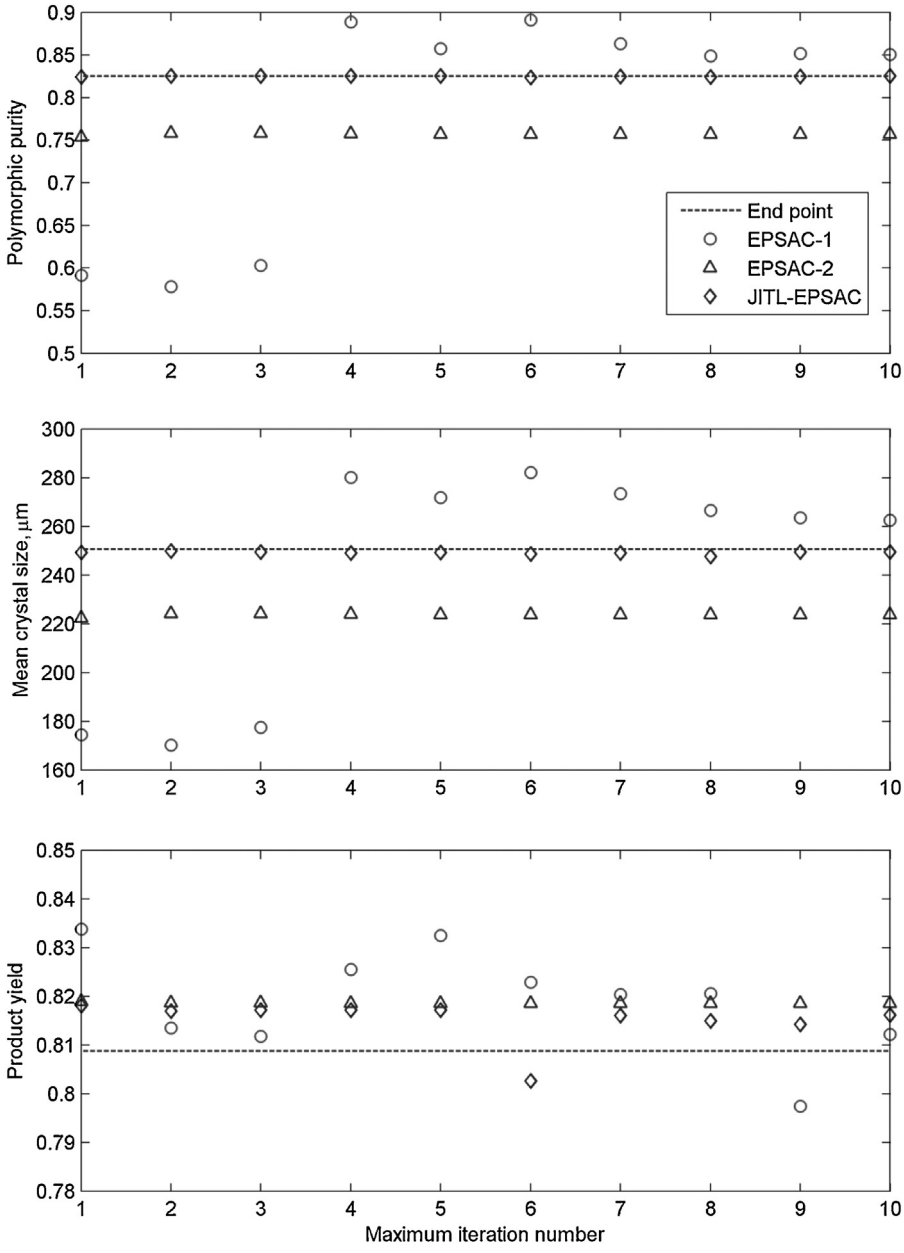

Fig. 5. Effect of maximum iteration number on the EPSAC and JITL-EPSAC closed-loop performance.

$$\vdots$$

$$\mathbf{x}_{l,\,k+N_u} = \left(\prod_{i=k}^{k+N_u-1} \mathbf{A}_i\right)\mathbf{x}_{l,\,k} + \left(\prod_{i=k+1}^{k+N_u-1} \mathbf{A}_i\right)\mathbf{B}_k\delta\mathbf{u}_k + \left(\prod_{i=k+2}^{k+N_u-1} \mathbf{A}_i\right)\mathbf{B}_{k+1}\delta\mathbf{u}_{k+1} + \ldots + \mathbf{B}_{k+N_u-1}\delta\mathbf{u}_{k+N_u-1}$$

$$\mathbf{x}_{l,\,k+N_p} = \left(\prod_{i=k}^{k+N_p-1} \mathbf{A}_i\right)\mathbf{x}_{l,\,k} + \left(\prod_{i=k+1}^{k+N_p-1} \mathbf{A}_i\right)\mathbf{B}_k\delta\mathbf{u}_k + \left(\prod_{i=k+2}^{k+N_p-1} \mathbf{A}_i\right)\mathbf{B}_{k+1}\delta\mathbf{u}_{k+1} + \ldots + \left(\prod_{i=k+N_u}^{k+N_p-1} \mathbf{A}_i\right)\mathbf{B}_{k+N_u-1}\delta\mathbf{u}_{k+N_u-1} + \ldots + \mathbf{B}_{k+N_p-1}\delta\mathbf{u}_{k+N_p-1}$$

Since $\mathbf{x}_{l,\,k} = \mathbf{0}$, and $\delta\mathbf{u}_{k+i} = \mathbf{0}$ for $i \geq N_u$, then

$$\vdots$$

$$\boldsymbol{X}_l = \boldsymbol{G}_l\delta U \tag{14}$$

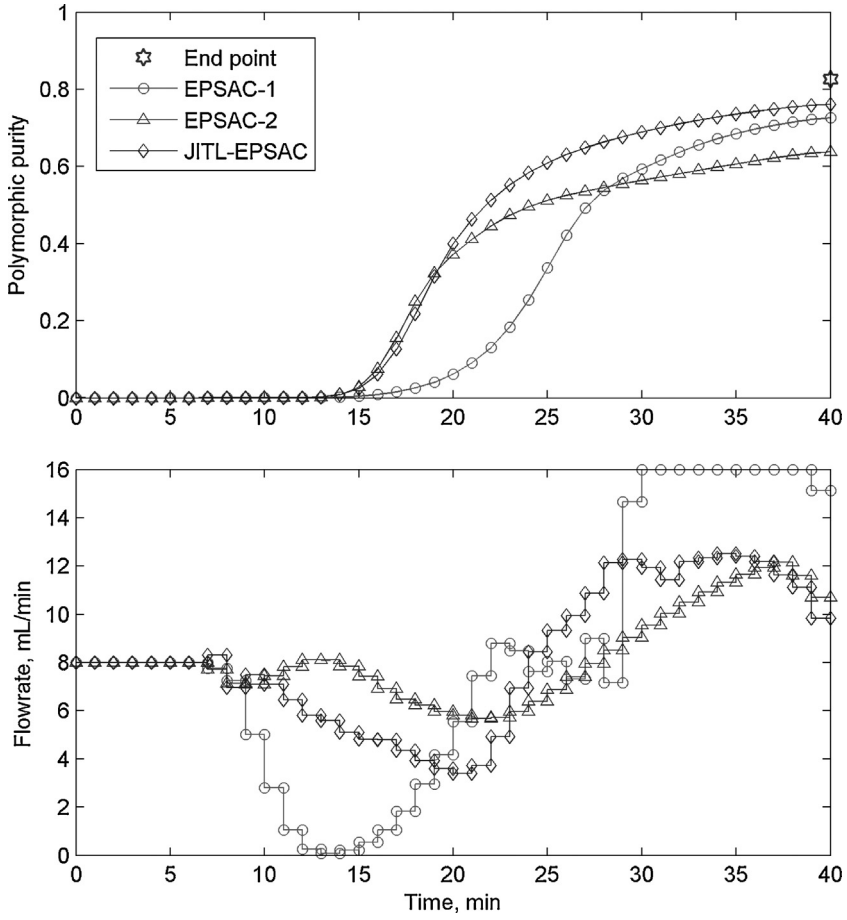

Fig. 6. Performance of batch-end property control by the EPSAC and JITL-EPSAC under model-plant mismatch of Case 1.

where $\mathbf{X}_l = \left[\mathbf{x}_{l,\,k+1}^T,\, \mathbf{x}_{l,\,k+2}^T,\, \cdots,\, \mathbf{x}_{l,\,k+N_p}^T\right]^T$, $\delta\mathbf{U} = \left[\delta\mathbf{u}_k^T,\, \delta\mathbf{u}_{k+1}^T,\, \cdots,\, \delta\mathbf{u}_{k+N_u-1}^T\right]^T$, and

$$\boldsymbol{G}_l = \begin{bmatrix} \boldsymbol{B}_k & \mathbf{0} & \cdots & \mathbf{0} \\ \boldsymbol{A}_{k+1}B_k & \boldsymbol{B}_{k+1} & \cdots & \mathbf{0} \\ \boldsymbol{A}_{k+2}A_{k+1}\boldsymbol{B}_k & \boldsymbol{A}_{k+2}\boldsymbol{B}_{k+1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \left(\prod_{i=k+1}^{k+N_u-1} A_i\right)B_k & \left(\prod_{i=k+2}^{k+N_u-1} A_i\right)B_{k+1} & \cdots & B_{k+N_u-1} \\ \vdots & \vdots & \ddots & \vdots \\ \left(\prod_{i=k+1}^{k+N_p-1} A_i\right)B_k & \left(\prod_{i=k+2}^{k+N_p-1} A_i\right)B_{k+1} & \cdots & \left(\prod_{i=k+N_u}^{k+N_p-1} A_i\right)B_{k+N_u-1} \end{bmatrix} \tag{15}$$

In summary, the future system state variables $\mathbf{X}$ in the control horizon can be conveniently represented in matrix form as

$$X = \boldsymbol{X}_b + \boldsymbol{G}_l\delta U \tag{16}$$

where $\mathbf{X} = \left[\mathbf{x}_{k+1}^T,\, \mathbf{x}_{k+2}^T, \ldots,\, \mathbf{x}_{k+N_p}^T\right]^T$, $\mathbf{X}_b = \left[\mathbf{x}_{b,\,k+1}^T,\, \mathbf{x}_{b,\,k+2}^T,\, \ldots,\, \mathbf{x}_{b,\,k+N_p}^T\right]^T$, and the control trajectory is defined as

$$\boldsymbol{U} = \boldsymbol{U}_b + \delta\boldsymbol{U} \tag{17}$$

where $\mathbf{U} = \left[\mathbf{u}_k^T,\, \mathbf{u}_{k+1}^T, \ldots,\, \mathbf{u}_{k+N_u-1}^T\right]^T$ and $\mathbf{U}_b = \left[\mathbf{u}_{b,\,k}^T,\, \mathbf{u}_{b,\,k+1}^T,\, \ldots,\, \mathbf{u}_{b,\,k+N_u-1}^T\right]^T$.

With Eqs. (16) and (17), the conventional routine for a quadratic cost function with nonlinear soft constraints could be straightforwardly implemented to obtain the optimal control moves of $\delta\mathbf{U}$. The detailed derivation and implementation procedure of the proposed control strategy are summarized as in Appendix A.

4. Results and discussion

The conventional EPSAC and the proposed JITL-EPSAC techniques are applied to a simulated semi-batch pH-shift reactive crystallization process of L-glutamic acid [29,30]. The crystallizer is initially filled with 0.65 L of monosodium glutamate (MSG) of 1.0 mol/L and the default batch time is 40 min. The manipulated variable (F, mL/min) is the addition flowrate of sulfuric acid (SA) of 1.0 mol/L, which is constrained between 0 and 16 mL/min while adjusted every minute to achieve the maximum polymorphic purity of α-form (P_α), volume-based mean crystal size (M_s, μm), and product yield of the final crystalline product at the batch end (P_y). The Pareto-optimality front for this multi-objective optimization was previously studied using a first-principles mathematical model [31,32]. The chosen optimal operating point is batch-end product quality [P_α, M_s, P_y] chosen to be [0.8255, 250.6, 0.8088].

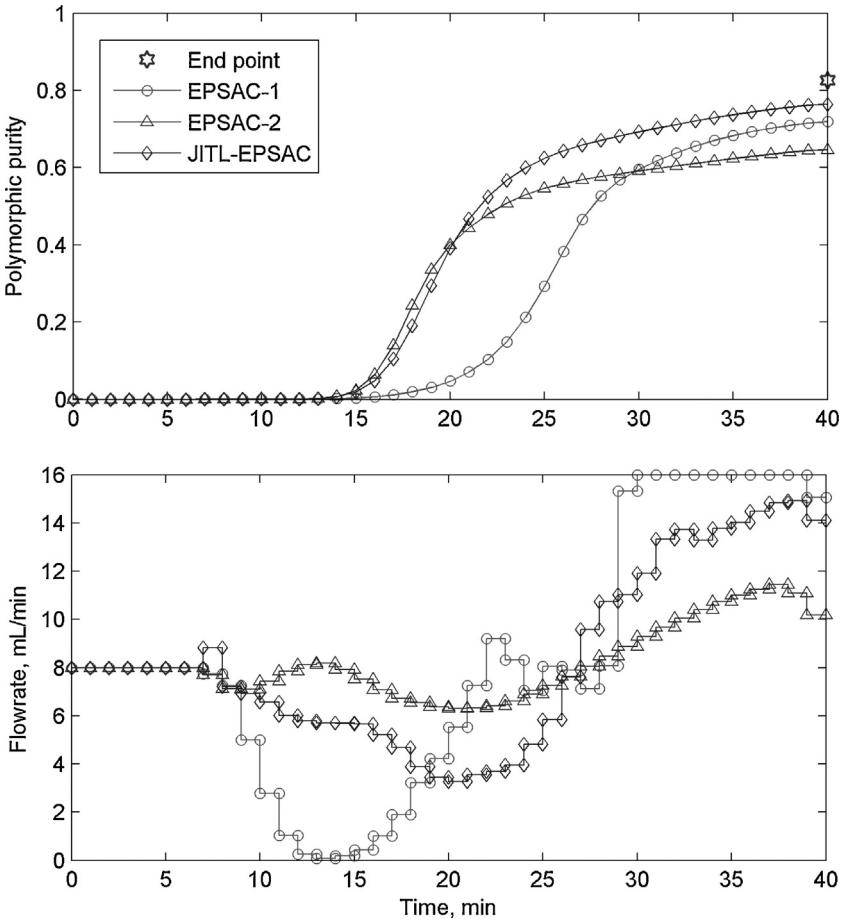

Fig. 7. Performance of batch-end property control by the EPSAC and JITL-EPSAC under model-plant mismatch of Case 2.

To proceed the JITL modeling of the crystallization process, the local state-space model is considered as

$$\begin{bmatrix} (P_\alpha)_{l,k+1} \\ (M_s)_{l,k+1} \\ (C)_{l,k+1} \\ (V)_{l,k+1} \end{bmatrix} = \boldsymbol{A}_k \begin{bmatrix} (P_\alpha)_{l,k} \\ (M_s)_{l,k} \\ (C)_{l,k} \\ (V)_{l,k} \end{bmatrix} + \boldsymbol{B}_k[\delta F_k] \tag{18}$$

where C (mol/L) is the glutamic acid concentration and V is the solution volume in the crystallizer with capacity of 0.97 L.

To generate the reference database for the JITL identification of local state-space models of Eq. (18), one hundred batches of process data were obtained by introducing step changes of random magnitude with $N(0, 1)$ to the constant addition flowrate of 8 mL/min as shown in Fig. 3. For the purpose of comparison, the EPSAC algorithm developed by Hermanto et al. [15] is also designed in the ensuing case studies. The proposed JITL-EPSAC algorithm differs from this previous EPSAC algorithm mainly due to the construction of $\mathbf{G}_{pl}$ and $\mathbf{G}_{hl}$ in Eqs. (15) and (A15), where the previous algorithm used a finite step response model to develop these two dynamic matrices. Both controllers use a prediction/control horizon starting from current time to the batch end, viz., shrinking horizon mode.

Both EPSAC controllers were initialized with a reference addition flowrate trajectory set as the nominal constant 8 mL/min. The control actions were activated only after measurements of polymorphic purity by Raman spectroscopy were steadily available [29,30]. The maximum iteration number set for both EPSAC algorithms was limited to one, viz., no iteration. The effect of increasing maximum iteration number will be discussed later. No model-plant mismatch was considered at this stage and all the state variables in the nonlinear process model were measureable or observable as studied by Mesbah et al. [33] and Hermanto et al. [15]. Tuning parameters $\mathbf{W}_p$ and $\mathbf{W}_u$ for product quality and control moves were specifically fine-tuned for conventional EPSAC as $1 \times 10^{-5}\,\mathbf{I}$ and unit matrix $\mathbf{I}$, respectively. These two weights were also used by the proposed JITL-EPSAC algorithm, so the difference in closed-loop performance between the two EPSAC algorithms are entirely due to the effect of process model employed.

The performance of the two EPSAC algorithms are shown by EPSAC-1 and JITL-EPSAC, respectively, in Fig. 4, where JITL-EPSAC could meet the control target with a smooth addition flowrate trajectory, while EPSAC-1 failed to meet the target and the input trajectory varied greatly in the later phase of the batch. As discussed in the introduction section, using a finite step response model established at current time to predict the controlled variable deep in the prediction horizon may become inaccurate for processes with considerable nonlinearity. One remedial to lessen this problem is to use adaptive weight $\mathbf{W}_u$ that penalizes control moves deep in the control horizon [15]. A similar weight was also implemented for successive linearization [3]. However, the tuning of such $\mathbf{W}_u$ is not a trivial task. A fine-tuned $\mathbf{W}_u$ that improves the batch-end property control with smooth control moves is

$$(\boldsymbol{W}_u)_{i,i} = 1 \times 10^{-5}[1 + 60(i-1)], \quad i = 1, \ldots, N_u \tag{19}$$

where $(\mathbf{W}_u)_{i,i}$ is the diagonal elements of matrix $\mathbf{W}_u$; and N_u is the control horizon. The corresponding closed-loop performance of EPSAC is shown by EPSAC-2 in Fig. 4.

As the main idea of EPSAC is to iteratively linearize along the reference trajectories, the maximum iteration number places an

important effect on the closed-loop performance of EPSAC. The influence of maximum iteration number on the convergence of the product quality variables is illustrated in Fig. 5, from which the proposed JITL-EPSAC algorithm shows fast convergence to the desired polymorphic purity P_α and mean crystal size M_s with slight variation observed for the product yield P_y for maximum iteration number specified from 1 to 10. For EPSAC-2, there is not much improvement by increasing the maximum iteration number due to the use of a conservative weight of $\mathbf{W}_u$ in Eq. (19), which strongly penalizes the control moves. EPSAC-1 steadily improves by increasing the maximum iteration number larger than 7. This case study shows that, if the finite step response model is not accurate for conventional EPSAC design, increasing the maximum iteration number only guarantee higher computational cost, but not necessarily to help improve the EPSAC performance.

As model-plant mismatch is inevitable in practice, to investigate the influence of parameter uncertainties on the performance of EPSAC algorithms, two case studies of kinetic uncertainties are introduced to the reactive crystallization process. Case 1 considers a 20% decrease of the α-form growth rate. Case 2 introduces 20% variations to four kinetic parameters studied in previous work [32], which are $(1+20\%)k_{b,\alpha 1}$, $(1-20\%)k_{g,\alpha 1}$, $(1+20\%)E_{g,\alpha 2}$, and $(1-20\%)k_{g,\beta 1}$. A maximum iteration number of 10, 2 and 2 were employed for EPSAC-1, EPSAC-2, and JITL-EPSAC, respectively. The results shown in Figs. 6 and 7 indicate that a much improved performance of JITL-EPSAC compared to the EPSAC algorithms under the model-plant mismatch demonstrates a high degree of robustness of the JITL-EPSAC algorithm.

5. Conclusion

The use of finite step/impulse response model for trajectory optimization in EPSAC algorithm is only accurate for linear or mildly nonlinear process. In order to enhance the accuracy of optimization along the reference trajectory for EPSAC applications in nonlinear processes, this article proposes the use of local state-space models identified by the JITL method. Local state-space models can well represent the high nonlinearity and the time-varying characteristic of chemical processes, particularly in batch/semi-batch processes. Easier weight tuning, smoother control moves, less computational burden, and better closed-loop performance of the proposed JITL-EPSAC algorithm over the conventional EPSAC algorithm for batch-end property control are demonstrated in a simulated semi-batch pH-shift reactive crystallization process.

Appendix A.

Consider an objective function J defined *by*

$$J = \min_{\Delta \boldsymbol{U}} [\boldsymbol{P} - \boldsymbol{P}_d]^T \boldsymbol{W}_p [\boldsymbol{P} - \boldsymbol{P}_d] + \Delta \boldsymbol{U}^T \boldsymbol{W}_u \Delta U \tag{A1}$$

where $\mathbf{P}$, $\mathbf{P}_d$, and $\Delta\mathbf{U}$ are the matrices of the interested product quality, desired product quality, and the change in input variables, respectively, given by

$$P = [\boldsymbol{P}_{k+1}^T,\ \boldsymbol{P}_{k+2}^T, \ldots,\ \boldsymbol{P}_{k+N_p}^T]^T \tag{A2}$$

$$\boldsymbol{P}_d = [\boldsymbol{P}_{d,\,k+1}^T,\ \boldsymbol{P}_{d,\,k+2}^T, \ldots,\ \boldsymbol{P}_{d,\,k+N_p}^T]^T \tag{A3}$$

$$\Delta U = [\boldsymbol{u}_k^T - \boldsymbol{u}_{k-1}^T,\ \boldsymbol{u}_{k+1}^T - \boldsymbol{u}_k^T, \ldots,\ \boldsymbol{u}_{k+N_u-1}^T - \boldsymbol{u}_{k+N_u-2}^T]^T \tag{A4}$$

and $\mathbf{W}_p$ and $\mathbf{W}_u$ are the weight matrices for the product quality and the change in input variables, respectively. Then $\mathbf{P}$ and $\Delta\mathbf{U}$ can be decomposed into

$$P = \boldsymbol{M}\boldsymbol{X} = \boldsymbol{M}(\boldsymbol{X}_b + \boldsymbol{G}_l \delta U) = \boldsymbol{M}\boldsymbol{X}_b + \boldsymbol{M}\boldsymbol{G}_l \delta \boldsymbol{U} \tag{A5}$$

where

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{m}_{k+1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{m}_{k+2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{m}_{k+N_p} \end{bmatrix} \tag{A6}$$

is designed to extract out the interested combination of product quality variables. When only the first state variable is selected,

$$\boldsymbol{m}_{k+1} = \boldsymbol{m}_{k+2} = \ldots = \boldsymbol{m}_{k+N_p} = [1,\ 0,\ 0,\ \ldots,\ 0]_{1\times n_x} \tag{A7}$$

Equation (A5) is further simplified as

$$P = \boldsymbol{P}_b + \boldsymbol{G}_{pl} \delta U \tag{A8}$$

where

$$\boldsymbol{P}_b = \mathrm{M}\boldsymbol{X}_b \tag{A9}$$

$$\boldsymbol{G}_{pl} = M\boldsymbol{G}_l \tag{A10}$$

where $\mathbf{G}_{pl}$ is the state-space model coefficient matrix corresponding to the product quality variable, $\mathbf{P}_b$ is the predicted product quality calculated using the nonlinear model with predetermined future inputs $\mathbf{U}_b = [\mathbf{u}_{b,\,k}^T,\ \mathbf{u}_{b,\,k+1}^T,\ \ldots,\ \mathbf{u}_{b,\,k+N_u-1}^T]^T$, and

$$\Delta \boldsymbol{U}_b = [\boldsymbol{u}_{b,\,k}^T - \boldsymbol{u}_{b,\ k-1}^T,\ \boldsymbol{u}_{b,\,k+1}^T - \boldsymbol{u}_{b,\,k}^T, \ldots,\ \boldsymbol{u}_{b,\,k+N_u-1}^T - \boldsymbol{u}_{b,\ k+N_u-2}^T]^T \tag{A11}$$

$$C = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ -\boldsymbol{I} & \boldsymbol{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & -\boldsymbol{I} & \boldsymbol{I} \end{bmatrix} \tag{A12}$$

$$\Delta \boldsymbol{U} = \Delta \boldsymbol{U}_b + \boldsymbol{C}\delta \boldsymbol{U} \tag{A13}$$

Hence, the minimization problem can be written as

$$\begin{aligned} J &= \min_{\delta \boldsymbol{U}} (\boldsymbol{P}_b + \boldsymbol{G}_{pl}\delta U - \boldsymbol{P}_d)^T W_p (\boldsymbol{P}_b + \boldsymbol{G}_{pl}\delta U - \boldsymbol{P}_d) \\ &\quad + (\Delta \boldsymbol{U}_b + \boldsymbol{C}\delta \boldsymbol{U})^T W_u (\Delta \boldsymbol{U}_b + \boldsymbol{C}\delta \boldsymbol{U}) \\ &= \min_{\delta U} \delta \boldsymbol{U}^T \boldsymbol{G}_{pl}^T \boldsymbol{W}_p \boldsymbol{G}_{pl} \delta U + 2(\boldsymbol{P}_b - \boldsymbol{P}_d)^T \boldsymbol{W}_p \boldsymbol{G}_{pl} \delta U \\ &\quad + \delta \boldsymbol{U}^T \boldsymbol{C}^T \boldsymbol{W}_u C \delta \boldsymbol{U} + 2\Delta U_b^T \boldsymbol{W}_u C \delta \boldsymbol{U} \\ &= \min_{\delta U} \delta \boldsymbol{U}^T \boldsymbol{\Gamma} \delta U + \boldsymbol{\Phi}^T \delta \boldsymbol{U} \end{aligned} \tag{A14}$$

where

$$\boldsymbol{\Gamma} = \mathbf{G}_{pl}^T \mathbf{W}_p \mathbf{G}_{pl} + \mathbf{C}^T \mathbf{W}_u \mathbf{C}$$

$$\boldsymbol{\Phi} = 2[(\mathbf{P}_b - \mathbf{P}_d)^T \mathbf{W}_p \mathbf{G}_{pl} + \Delta \mathbf{U}_b^T \mathbf{W}_u \mathbf{C}]^T$$

The minimization is subject to the constraint $h(\mathbf{x}_j, \mathbf{u}_j) \le \mathbf{0}$, $\forall j \ge k$, where k is the current sampling instance. For notational convenience, $h\left(\mathbf{x}_j, \mathbf{u}_j\right) \le \mathbf{0}$ is denoted as $\mathbf{h}_j$, which can be decomposed into the base and linear part $\mathbf{h}_j = \mathbf{h}_{b,j} + \mathbf{h}_{l,j}$. Therefore, the matrix form of the constraints in the control horizon is

$$\boldsymbol{H}_b + \boldsymbol{G}_{hl}\delta U \le \mathbf{0} \tag{A15}$$

where $\mathbf{G}_{hl}$ is the state-space model coefficient matrix corresponding to the constraints function $\mathbf{h}_j$ and $\mathbf{H}_b = [\mathbf{h}_{b,k}^T, \ldots, \mathbf{h}_{b,k+N_p}^T]^T$.

When the constraints are highly nonlinear, handling (A15) directly will sometimes cause difficulty for the quadratic programming (QP) used for the optimization to find a feasible solution.

Convergence was provided by the soft constraint approach, which replaces the minimization problem with

$$\min_{\delta \boldsymbol{U}, \varepsilon} J_{sc} \tag{A16}$$

subject to

$$\boldsymbol{H}_b + \boldsymbol{G}_{hl}\delta U \le \boldsymbol{\varepsilon} \tag{A17}$$

$$\boldsymbol{\varepsilon} \ge \boldsymbol{0} \tag{A18}$$

where $J_{sc} = J + \boldsymbol{\varepsilon}^T \mathbf{W}_\varepsilon \boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^T \mathbf{w}_\varepsilon$, $\boldsymbol{\varepsilon}$ is a vector of slack variables, $\mathbf{W}_\varepsilon$ is a diagonal matrix of positive weight, and $\mathbf{w}_\varepsilon$ is a vector of positive elements. This modified minimization problem can be written as

$$\begin{aligned} J^*_{sc} = & \min_{\delta U, \varepsilon} \delta \boldsymbol{U}^T \boldsymbol{\Gamma} \delta \boldsymbol{U} + \boldsymbol{\Phi}^T \delta \boldsymbol{U} + \boldsymbol{\varepsilon}^T W_\varepsilon \boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^T \boldsymbol{w}_\varepsilon \\ = & \min_{\delta U, \varepsilon} [\ \delta \boldsymbol{U}^T \quad \boldsymbol{\varepsilon}^T\ ] \begin{bmatrix} \Gamma & 0 \\ 0 & \boldsymbol{W}_\varepsilon \end{bmatrix} \begin{bmatrix} \delta \boldsymbol{U} \\ \boldsymbol{\varepsilon} \end{bmatrix} + [\ \Phi^T \quad w_\varepsilon^T\ ] \begin{bmatrix} \delta \boldsymbol{U} \\ \boldsymbol{\varepsilon} \end{bmatrix} \\ = & \min_{\boldsymbol{\Pi}} \ \boldsymbol{\Pi}^T \Lambda \boldsymbol{\Pi} + \boldsymbol{\tau}^T \boldsymbol{\Pi} \end{aligned} \tag{A19}$$

subject to

$$\begin{bmatrix} \boldsymbol{H}_b \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{G}_{hl} & -I \\ 0 & -\boldsymbol{I} \end{bmatrix} \boldsymbol{\Pi} \le \boldsymbol{0} \tag{A20}$$

where $\boldsymbol{\Pi} = [\ \delta \mathbf{U}^T \quad \boldsymbol{\varepsilon}^T\ ]^T$, $\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Gamma} & 0 \\ 0 & \mathbf{W}_{\boldsymbol{\varepsilon}} \end{bmatrix}$, and $\boldsymbol{\tau} = [\ \Phi^T \quad \mathbf{w}_{\boldsymbol{\varepsilon}}^T\ ]^T$.

To summarize, the procedure for implementing the ESPAC strategy based on JITL-based local models at each sampling instant k is:

(1) Obtain $\mathbf{U}_b$ by

- If $k = 0$ and $iter = 1$, $\mathbf{U}_b$ is specified to be the control actions implemented in the previous batch;
- If $k > 0$ and $iter = 1$, $\mathbf{U}_b$ is set as the $\mathbf{U}_{optimal}$ obtained at the previous sampling instant of the current batch, where $iter$ is the iteration count;
- If $k > 0$ and $iter > 1$, the updated $\mathbf{U}_b$ from the previous iteration is used.

(2) Given the predicted state variables, obtain $\mathbf{P}_b$ and $\mathbf{H}_b$ by using $\mathbf{U}_b$ as the input to the nonlinear process model (1)–(5).
(3) Obtain the state-space model coefficient matrices $\mathbf{G}_{pl}$ and $\mathbf{G}_{hl}$ by using JITL with reference to the query point.
(4) Obtain $\boldsymbol{\Pi}^* = [\ \delta \mathbf{U}^{*T} \quad \boldsymbol{\varepsilon}^{*T}\ ]^T$ from the solution to the minimization problem (A19) and (A20), then update the element of U_b using

$$\mathbf{u}_{b,\,k+j} = \mathbf{u}_{b,\,k+j} + \delta \mathbf{u}_{k+j}$$

where $j = 0, \ldots, N_u - 1$.

(5) Calculate $err = \| [\ \begin{matrix} \mathbf{G}_{pl} \\ \mathbf{G}_{hl} \end{matrix}\ ] \delta \mathbf{U}^* \|$. If err is greater than a specified tolerance, $iter = iter + 1$, and go back to Step (1). Otherwise, set $\mathbf{U}_{optimal} = \mathbf{U}_b$ and implemented the first element of $\mathbf{U}_{optimal}$ to the process.

References

[1] F. Manenti, Considerations on nonlinear model predictive control techniques, Comput. Chem. Eng. 35 (2011) 2491–2509.
[2] M.L. Darby, M. Nikolaou, MPC: current practice and challenges, Control Eng. Pract. 20 (2012) 328–342.
[3] J.H. Lee, N.L. Ricker, Extended Kalman filter based nonlinear model predictive control, Ind. Eng. Chem. Res. 33 (1994) 1530–1541.
[4] H. Peng, K. Nakano, H. Shioya, Nonlinear predictive control using neual nets-based local linearization ARX model-stability and industrial application, IEEE Trans. Control Syst. Technol. 15 (2007) 130–143.
[5] Z.K. Nagy, R.D. Braatz, Robust nonlinear model predictive control of batch processes, AIChE J. 49 (2003) 1776–1786.
[6] Z.K. Nagy, F. Allgöwer, A nonlinear model predictive control approach for robust end-point property control of a thin-film deposition process, Int. Robust Nonlinear Control 17 (2007) 1600–1613.
[7] L. Özkan, M.V. Kothare, C. Georgakis, Model predictive control of nonlinear systems using piecewise linear models, Comput. Chem. Eng. 24 (2000) 793–799.
[8] A.L. Cervantes, O.E. Agamennoni, J.L. Figueroa, A nonlinear model predictive control system based on Wiener piecewise linear models, J. Process Control 13 (2003) 655–666.
[9] S. García-Nieto, M. Martínez, X. Blasco, J. Sanchis, Nonlinear predictive control based on local model networks for air management in diesel engines, Control Eng. Pract. 16 (2008) 1399–1413.
[10] M. Kuure-Kinsey, B.W. Bequette, A multiple model predictive control strategy for disturbance rejection, Ind. Eng. Chem. Res. 49 (2010) 7983–7989.
[11] M.W. Hermanto, R.D. Braatz, M.S. Chiu, Integrated batch-to-batch and nonlinear model predictive control for polymorphic transformation in pharmaceutical crystallization, AIChE J. 57 (2011) 1008–1019.
[12] R.M. De Keyser, A.R. Cauwenberghe, Extended prediction self-adaptive control, in: Paper Presented At: The IFAC Symposium on Identification and System Parameter Estimation, York, UK, 1985.
[13] R. De Keyser, Model based predictive control for linear systems, in: H. Unbehauen (Ed.), UNESCO Encyclopaedia of Life Support System (EoLSS), vol. XI, Eolss Publishers Co., Ltd., Oxford, 2003.
[14] M. Gálvez-Garrillo, R. De Keyser, C. Ionescu, Nonlinear predictive control with dead-time compensator: application to a solar power plant, Sol. Energy 83 (2009) 743–752.
[15] M.W. Hermanto, M.S. Chiu, R.D. Braatz, Nonlinear model predictive control for the polymorphic transformation of L-glutamic acid crystals, AIChE J. 55 (2009) 2631–2645.
[16] A. Dutta, Y. Zhong, B. Depraetere, et al., Model-based and model-free learning strategies for wet clutch control, Mechatronics 24 (8) (2014) 1008–1020.
[17] I. Nascu, A. Krieger, C.M. Ionescu, E.N. Pistikopoulos, Advanced model-based control studies for the induction and maintenance of intravenous anaesthesia, IEEE Trans. Biomed. Eng. 62 (3) (2015) 832–841.
[18] C. Cheng, M.S. Chiu, A new data-based methodology for nonlinear process modeling, Chem. Eng. Sci. 59 (2004) 2801–2810.
[19] Q. Su, M. Kano, M.S. Chiu, A new strategy of locality enhancement for Just-in-Time Learning method, Comput. Aided Chem. Eng. 31 (2012) 1662–1666.
[20] P.O. Scokaert, J.B. Rawlings, Feasibility issues in linear model predictive control, AIChE J. 45 (1999) 1649–1659.
[21] A. Rueda, S. Cristea, C.D. Prada, R.M. De Keyser, Non-linear predictive control for a distillation column, in: Paper Presented at: The 44th IEEE Conference on Decision and Control, and the European Control Conference, Seville, Spain, 2005.
[22] B.A. Foss, T.A. Johansen, A.V. Sørensen, Nonlinear predictive control using local models-applied to a batch fermentation process, Control Eng. Pract. 3 (1995) 389–396.
[23] K.P. Dharaskar, Y.P. Gupta, Predictive control of nonlinear processes using interpolated models, Chem. Eng. Res. Des. 78 (2000) 573–580.
[24] K. Fujiawara, M. Kano, S. Hasebe, A. Takinami, Soft-sensor development using correlation-based Just-in-Time modeling, AIChE J. 55 (2009) 1754–1765.
[25] C. Cheng, M.S. Chiu, Nonlinear process monitoring using JITL-PCA, Chemom. Intell. Lab. Syst. 76 (2005) 1–13.
[26] M. Kano, T. Sakata, S. Hasebe, Just-In-Time statistical process control: adaptive monitoring of vinyl acetate monomer process, in: Paper Presented at: The 18th IFAC World Congress, Milano, Italy, 2011.
[27] X. Yang, L. Jia, M.S. Chiu, Adaptive decentralized PID controllers design using JITL modeling methodology, J. Process Control 22 (2012) 1531–1542.
[28] Q.L. Su, R.D. Braatz, M.S. Chiu, JITL-based concentration control for semi-batch pH-shift reactive crystallization of L-glutamic acid, J. Process Control 24 (2014) 415–421.
[29] H. Alatalo, J. Kohonen, H. Qu, et al., In-line monitoring of reactive crystallization process based on ATR-FTIR and Raman spectroscopy, J. Chemom. 22 (2008) 644–652.
[30] H. Qu, H. Alatalo, H. Hatakka, M. Louhi-Kultanen, S.P. Reinikainen, J. Kallas, Raman and ATR FTIR spectroscopy in reactive crystallization: simultaneous monitoring of solute concentration and polymorphic state of the crystals, J. Cryst. Growth 311 (2009) 3466–3475.
[31] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2002) 182–197.
[32] Q.L. Su, R.D. Braatz, M.S. Chiu, Modeling and Bayesian parameter estimation for semi-batch pH-shift reactive crystallization of L-glutamic acid, AIChE J. 60 (2014) 2828–2838.
[33] A. Mesbah, A.E. Huesman, H.J. Kramer, Z.K. Nagy, P.M. Van de Hof, Real-time control of a semi-industrial fed-batch evaporative crystallizer using different direct optimization strategies, AIChE J. 57 (2011) 1557–1569.