Batch-to-batch Strategies for Cooling Crystallization

Marco Forgione, Ali Mesbah, Xavier Bombois and Paul M.J. Van den Hof

Abstract— **Two batch-to-batch (B2B) algorithms for super-saturation control in cooling crystallization are presented in this paper. In Iterative Learning Control (ILC) a nominal process model is adjusted with an additive correction term which depends on the error in the last batch. In Iterative Identification Control (IIC) the physical parameters of the process model are recursively estimated by adopting a Bayesian identification framework. Both B2B algorithms compute an optimized input for the next batch that is fed to a lower level PI feedback controller in order to reject the process disturbances. The tracking performance of these B2B+PI control schemes is investigated in a simulation study.**

I. Introduction

Batch crystallization is a separation process extensively utilized in the pharmaceutical, food and fine chemical sectors for the production of high value-added specialty chemicals. The most common way to operate industrial batch cooling crystallizers is to control the temperature inside the crystallizer in order to follow a desired profile during the batch. This strategy is known as T-control in literature [3]. Since accurate on-line temperature measurements can readily be obtained, the temperature is usually controlled in a closed-loop setting. In this configuration, the desired temperature is given as reference to a feedback loop.

However, even when the temperature is effectively controlled, the final product of a batch might not show the expected characteristics. The temperature is an important process variable, but it is not the one most closely related to the crystallization dynamics. The variable having the most direct influence on the physico-chemical phenomena occurring in crystallization is the supersaturation. This last quantiy is often defined in terms of the solute concentration.

Feedback control strategies for supersaturation tracking have been widely investigated [6], [7]. They are known as C-control strategies in literature [3]. In general, C-control was shown to give better performance compared to T-control, particularly in terms of reproducibility of the final product. A condition for the implementation of feedback C-control is that accurate and reliable on-line concentration measurements are available. However, in some cases the measurements are obtained from off-line analysis of samples collected throughout the batch. In other cases, the measurement collected on-line are not considered reliable enough for on-line control, especially in an industrial environment.

In such situations feedback C-control is not feasible, but we can still achieve supersaturation tracking by using the information batchwise.

In this paper, we present a Batch-to-Batch (B2B) control strategy to track efficiently a supersaturation profile under the presence of unknown disturbances and model uncertainties. Based on the desired supersaturation profile and the off-line concentration measurements from the previous batches, a B2B algorithm computes after each batch an improved reference profile T^r for the temperature in the crystallizer. This profile is set as reference for the lower level PI temperature controller in the next batch.

Two B2B algorithms, namely an Iterative Learning Control (ILC) and an Iterative Identification Control (IIC) are presented and compared. The ILC algorithm has already been introduced in [2]. In this paper, the tuning of ILC is made easier by posing its derivation in a stochastic settings inspired from [1]. The IIC algorithm is based on System Identification theory and is the main contribution of this paper. Both algorithms update a model estimate after each batch based on the newest temperature and concentration measurements. Subsequently, they compute the input for the next batch by solving an optimization problem based on the most recent model estimate in order to track the desired supersaturation reference.

However, the nature of the model update is different in the two cases. In ILC the nominal process model is adjusted with an additive correction term which depends on the model error in the last iteration. On the contrary, in IIC the physical parameters of the process model are recursively estimated adopting a probabilistic Bayesian framework in such a way that the estimate is improved from batch to batch. Advantages and disadvantages of the two methods are investigated in a simulation study.

II. Model of Batch Cooling Crystallization

The dynamics of a crystallization process is usually described using a *Population Balance Equation* (PBE), along with conservation balance equations and kinetic relations [9]. The PBE is a partial differential equation which describes the time evolution of the *Crystal Size Distribution* (CSD), i.e. the distribution of the number of crystals in the size domain. The overall model is composed of the PBE and a set of differential, possibly algebraic differential equations.

Under certain assumptions on the kinetic phenomena, a model reduction technique known as *moment model reduction* can be applied [9]. This reduction leads to a set of ordinary differential equations known as *moment equations*. The process model used in this paper is based on moment

equations and its derivation is explained in details in [2]. Due to space limitation, here we only report a state-space representation of this model in the form

$$\begin{aligned}
\dot{x} &= \mathcal{F}(x(t)) + \mathcal{G}(T_J(t)) \\
y &= \mathcal{H}(x(t)) \\
S &= \mathcal{M}(x(t)).
\end{aligned} \tag{1}$$

The states $x = (m_0 \ m_1 \ m_2 \ m_3 \ T)^\top$ are the first four moments of the CSD and the crystallizer temperature, the input is the jacket temperature T_J, the measured output $y = (T \ C)^\top$ are the crystallizer temperature and the solute concentration, the control output S is the supersaturation. The state and the output mappings are given by

$$\mathcal{F}(x) = \begin{pmatrix} k_b\big(C_0 - 10^{-3}\rho_c k_v(m_3 - m_{3,0}) - C_s(T)\big)^b m_3 \\ k_g\big(C_0 - 10^{-3}\rho_c k_v(m_3 - m_{3,0}) - C_s(T)\big)^g m_0 \\ 2k_g\big(C_0 - 10^{-3}\rho_c k_v(m_3 - m_{3,0}) - C_s(T)\big)^g m_1 \\ 3k_g\big(C_0 - 10^{-3}\rho_c k_v(m_3 - m_{3,0}) - C_s(T)\big)^g m_2 \\ -\frac{UA}{\rho c_p V}T \end{pmatrix}$$

$$\mathcal{G}(T_J) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{UA}{\rho c_p V}T_J \end{pmatrix} \quad \mathcal{H} = \begin{pmatrix} T \\ C_0 - 10^{-3}\rho_c k_v(m_3 - m_{3,0}) \end{pmatrix}$$

$$\mathcal{M} = C_0 - 10^{-3}\rho_c k_v(m_3 - m_{3,0}) - C_s(T) \tag{2}$$

where $C_s(T)$ is a static mapping representing the solubility. Note that the control output S can be computed from the measured outputs y using the static relation

$$S = C - C_s(T). \tag{3}$$

The kinetic parameters $\theta = \begin{bmatrix} k_b & b & k_g & g \end{bmatrix}^\top$ in the state equation may not be known with great accuracy in practice. We assume a nominal value $\hat{\theta}$ is available a priori. However, the actual parameters θ_0 used in the simulation model will differ from $\hat{\theta}$ in order to take the effect of parametric model mismatches into account. On the contrary, the other coefficients appearing in the model equations are assumed to be known exactly. The value of these coefficients, the nominal value of the parameters $\hat{\theta}$ and the initial condition of the model can be found in [2]. It is worthwhile noting that the process dynamics is split into a part from T_J to T that is perfectly known and a part from T to C that depends on the uncertain parameters θ (Figure 1, Batch Crystallizer block).

It will be convenient to consider the system in a finite discrete-time representation. By applying an integration method with fixed step $t_d = 5$ sec for a given initial condition the input/output relation from T_J to S is represented by the static mapping $\mathbf{S} = F_{ST_J}(\mathbf{T_J}, \theta)$. In the following, we shall adopt the bold-face notation for vectors of sampled variables $\in \mathbb{R}^N$, where N is the batch length. The notation $F_{WV}(\cdot)$ will be used for the mapping from an input vector $\mathbf{V}$ to an output vector $\mathbf{W}$.

Measurements $\tilde{C}$ and $\tilde{T}$ of C and T are collected at the same rate $t_s = t_d$ and corrupted by additive measurement
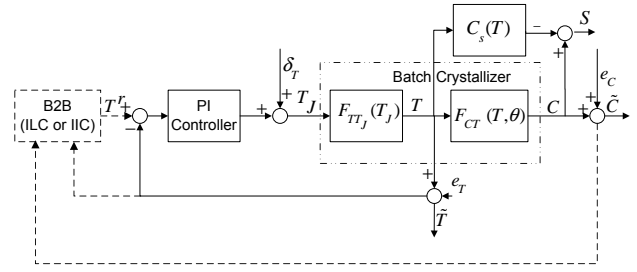

Fig. 1.   The overall B2B+PI control scheme

noise sequences e_C and e_T, respectively. e_C and e_T are modeled as realizations of independent white gaussian variables with standard deviation $\sigma_T = 0.1$ ℃ and $\sigma_C = 2$ g/L.

The jacket temperature T_J is perturbed by an additive low-frequency disturbance δ_T, which is modeled as an autoregressive process of order 1 with standard deviation $\sigma_{AR} = 0.25$ ℃:

$$\delta_T(t+1) = a\delta_T(t) + e(t) \tag{4}$$

where $a = 0.9895$ and $e(t)$ is white noise with standard deviation $\sigma_e = \sigma_{AR}\sqrt{1 - a^2}$.

The realizations e_C, e_T and δ_T are different for each batch.

III. BATCH-TO-BATCH SUPERSATURATION CONTROL

In this section we present the B2B strategies ILC and IIC. Both strategies are based on the same B2B+PI configuration that is presented in the first subsection. Subsequently, the role of the B2B controller is discussed, followed by a detailed description of the two algorithms.

A. B2B+PI Configuration

Due to the presence of disturbances on the temperature dynamics, we include into the control scheme a PI controller for the crystallizer temperature T. The B2B algorithm drives the set-point of the PI controller T^r instead of the jacket temperature T_J directly. The overall B2B+PI control scheme is sketched in Figure 1. The two leftmost blocks represent the control system: the PI temperature controller and the B2B controller, which drives the reference of the latter. The signals coming and departing from the B2B block are updated off-line only, i.e. from one batch to the other and are indicated by dashed lines. All other signals represented by continuous lines are updated during the same batch.

The PI controller is determined by the transfer function $C(s) = K_P + \frac{K_I}{s}$ where $K_P = (\rho c_p V)/(t_{cl}UA)$ and $K_I = 1/t_{cl}$ with $t_{cl} = 2$ min. A discrete time version of this PI controller is implemented in the simulation model.

B. Batch-to-Batch Control

After each batch, the corrupted measurements outputs $\tilde{\mathbf{y}} = (\tilde{\mathbf{T}}, \ \tilde{\mathbf{C}})^\top$ are available. The corrupted control output $\tilde{\mathbf{S}}$ can be estimated from $\tilde{\mathbf{y}}$ according to Equation (3). The role of the B2B controller is to design after each batch k an improved input $\mathbf{T}^r_{k+1}$ in order to track a reference $\bar{\mathbf{S}}_{k+1}$ in the batch $k + 1$. The design can be based on all the information collected up to batch k, that is given

by the data collected from the previous batches and the a priori information about the system. In our case the a priori information consists of an assumed *model structure* $\mathbf{S} = F_{ST^r}(\mathbf{T}^r, \theta)$, the *nominal parameter* $\hat{\theta}$ and the statistical properties of the disturbances acting on the system.

In general, *real dynamics* from T^r to S will differ from the *nominal model* $\hat{F}_{ST^r}(\mathbf{T}^r) \triangleq F_{ST^r}(\mathbf{T}^r, \hat{\theta})$ since $\hat{\theta} \neq \theta_0$ (parametric model mismatch). Furthermore, real dynamics might even not be exactly described by any parameter θ in the case of a structural model mismatches.

C. Iterative Learning Control

Different B2B control algorithms are chategorized as ILC in literature. In general, they are mappings defining the input in the next batch based on the nominal model and the previous input-output data.

The ILC algorithm we presented in [2] was based on an additive model correction. We defined the updated model at batch k as

$$\mathbf{S} = \hat{F}_{ST^r}(\mathbf{T}^r) + \boldsymbol{\alpha}_k \tag{5}$$

where $\boldsymbol{\alpha}_k$ is the so-called correction vector. The correction vector was estimated based on the measurement of the previous batch as

$$\boldsymbol{\alpha}_k = \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \|\tilde{\mathbf{S}}_k - (\hat{F}_{ST^r}(\mathbf{T}_k^r) + \boldsymbol{\alpha})\|^2 + S_\alpha \|\boldsymbol{\alpha} - \boldsymbol{\alpha}_{k-1}\|^2 \tag{6}$$

where S_α is a tuning parameter. However, the tuning of S_α was not intuitive and required intensive trial-and-error.

In this paper we introduce a novel framework in order to estimate the correction vector based on a stochastic setting inspired from [1]. This framework is intended to make the tuning of the algorithm easier and more efficient. The true system is assumed to satisfy the following stochastic model

$$\begin{aligned} \boldsymbol{\alpha}_k &= \boldsymbol{\alpha}_{k-1} + \boldsymbol{\Delta\alpha}_k, & \boldsymbol{\Delta\alpha}_k &\sim \mathcal{N}(0, \Sigma_{\alpha_k}) \\ \tilde{\mathbf{S}}_k &= \hat{F}_{ST^r}(\mathbf{T}_k^r) + \boldsymbol{\alpha}_k + \mathbf{v}_k, & \mathbf{v}_k &\sim \mathcal{N}(0, \Sigma_{v_k}) \end{aligned} \tag{7}$$

with $\boldsymbol{\alpha}_0 \sim \mathcal{N}(0, \Sigma_{\alpha_0})$.

The value $\boldsymbol{\alpha}_{k|k}$ can be estimated from the measurement $\tilde{\mathbf{S}}_k$ using the Kalman filter on the previous model. The correction vector $\boldsymbol{\alpha}_k$ models the deviation of the nominal output that will occur again at iteration $k+1$ modified by up to an innovation term $\boldsymbol{\Delta\alpha}_k$. This deviation might be due to the bias of the nominal model along a particular system trajectory, as well as due to repetitive disturbances acting on the system. On the contrary, the vector $\mathbf{v}_k$ models output deviations that will not be repeated at the next iteration. The latter might be due to measurement noise as well as due to the effect of nonrepetitive disturbances. A large Σ_{α_k} will force $\boldsymbol{\alpha}_k$ to adapt fast to $\tilde{\mathbf{S}}_k - \hat{F}_{ST^r}(\mathbf{T}_k^r)$ while a large Σ_{v_k} would make the adaptation slow since $\tilde{\mathbf{S}}_k$ would be considered less reliable. Furthermore, the expected frequency content of $\boldsymbol{\Delta\alpha}_k$ and $\mathbf{v}_k$ can be modeled in this framework leading to a more efficient filtering.

Given a linear filter $H(q)$, let us define $\mathcal{Q}_H^N \in \mathbb{R}^{N \times N}$ as the Toeplitz matrix with entries $\mathcal{Q}_H^N(i,j) = \mathcal{R}(i-j)/\mathcal{R}(0)$ where $\mathcal{R}(\tau)$ is the autocorrelation of a stochastic process

$w(t) = H(q)e(t)$ and $e(t)$ is white noise. $\mathcal{Q}_H^N$ is the covariance matrix of a vector $\mathbf{w}$ of random variables extracted from the stochastic process $w(t)$ and normalized to have unitary variance. We can thus specify Σ_{α_k} and Σ_{v_k} in terms of linear filters describing the frequency content of the corresponding signals using the operator $\mathcal{Q}$.

Compared to [1], the stochastic model is used to describe the evolution of the correction vector $\boldsymbol{\alpha}_k$ instead of the tracking error. Our formulation is therefore more flexible for the situations in which the set-point is allowed to change from one batch to another. Furthermore, a general nonlinear process dynamics is assumed instead of a linear one.

The covariance matrix Σ_{α_k} is set to $a_k^2 \mathcal{Q}_B^N$ where $B(q)$ is a fourth order Butterworth filter with cutoff frequency $1/t_f$, $t_f = 5$ minutes and a_k is an iteration-dependent scalar. The cutoff frequency was chosen by examining the usual frequency content of the correction vector during noise-free simulations. The iteration-dependent a_k is a free tuning parameter and represents the expected amplitude of the variation of the correction vector at iteration k. $\boldsymbol{\alpha}_k$ can be lowered after some iterations in order to increase the filtering effect on the estimate of the correction vector. However, after a set-point change the value should be increased again in order to let the correction vector adapt to the new configuration.

Σ_{v_k} is set to $\sigma_C^2 \mathcal{Q}_1^N = \sigma_C^2 I^N$ where I^N is the identity matrix in order to model the measurement noise on $\tilde{\mathbf{C}}$ used to estimate $\tilde{\mathbf{S}}$. The measurement noise on $\tilde{\mathbf{T}}$ and the nonlinear effect of the disturbances δ_T and e_T on $\tilde{\mathbf{C}}$ are ignored in this setting.

Based on the previous considerations, this algorithm has been devised. At each batch k, these steps are executed:

1) The temperature reference $\mathbf{T}_k^r$ is set as the input to the temperature controller and the noisy measurements $\tilde{\mathbf{C}}_k, \tilde{\mathbf{T}}_k$ are collected.
2) $\tilde{\mathbf{S}}_k$ is computed as $\tilde{\mathbf{C}}_k - C_s(\tilde{\mathbf{T}}_k)$.
3) A corrected model of the dynamics from T^r to S is found as $\mathbf{S} = \hat{S}_k(\mathbf{T}^r) = F_{ST^r}(\mathbf{T}^r; \hat{\theta}) + \boldsymbol{\alpha}_{k|k}$. The correction vector $\boldsymbol{\alpha}_{k|k}$ is estimated using the Kalman filter on the stochastic model

$$\begin{aligned} \boldsymbol{\alpha}_k &= \boldsymbol{\alpha}_{k-1} + \boldsymbol{\Delta\alpha}_k, \ \boldsymbol{\Delta\alpha}_k \sim \mathcal{N}(0, a_k^2 \mathcal{Q}_B^N) \\ \tilde{\mathbf{S}}_k &= \hat{F}_{ST^r}(\mathbf{T}^r) + \boldsymbol{\alpha}_k + \mathbf{e}_C, \ \mathbf{e}_C \sim \mathcal{N}(0, \sigma_C^2 I^N) \end{aligned} \tag{8}$$

where a_k is a free iteration-depent parameter.
4) The corrected model is used to compute the temperature profile for the next iteration

$$\mathbf{T}_{k+1}^r = \arg \min_{\mathbf{T}^r \in \mathbb{R}^N} \|\overline{\mathbf{S}}_{k+1} - \hat{S}_k(\mathbf{T}^r)\|^2 \tag{9}$$

where $\overline{\mathbf{S}}_{k+1}$ is the supersaturaton set-point for the batch $k+1$.

The dynamic optimization problem (9) is solved numerically using the active-set method of the Matlab function `fmincon`. The optimization is based on a *single shooting* strategy that is discussed in details in [2].

D. Iterative Identification Control

ILC algorithms are generally based on a nominal model, but not directly on the knowledge of a full model structure

parametrized in θ. Intuitively, the explicit use of this information may lead to a more efficient B2B learning algorithm.

The Bayesian framework suggests a recursive approach in order to update the model parameters based on the estimates from the previous batch and the data $\tilde{\mathbf{y}}_k$ measured in the current batch. After a batch k, the a posteriori probability distribution $p_{\theta|k}(\theta|k)$ of the model parameters can be computed according to Bayes rules using the previous parameter distribution $p_{\theta|k-1}(\theta|k-1)$ and the likelihood $p_{\tilde{\mathbf{y}}_k|\theta}(\tilde{\mathbf{y}}_k|\theta)$ of the measured output $\tilde{\mathbf{y}}_k$ from the last batch

$$p_{\theta|k}(\theta|k) = \frac{p_{\tilde{\mathbf{y}}_k|\theta}(\tilde{\mathbf{y}}_k|\theta)p_{\theta|k-1}(\theta|k-1)}{\int p_{\tilde{\mathbf{y}}_k|k}(\tilde{\mathbf{y}}_k|k)p_{\theta|k-1}(\theta|k-1)\,d\theta}. \quad (10)$$

The parameter estimates actually used to describe the system at iteration k could be chosen adopting different criteria, for instance the Maximum over θ of the A Posteriori distribution (MAP estimate). However, this estimation problem is in general complex and does not admit a close-form solution.

An analytically tractable case is given when the observation $\tilde{\mathbf{y}}_k$ is a vector of Gaussian variables with known covariance and mean value parametrized by θ, and the previous parameter distribution is also Gaussian i.e. $\tilde{\mathbf{y}}_k \sim \mathcal{N}(m(\theta_0), \Sigma_e)$ and $p_{\theta|k-1}(\theta|k-1) = \mathcal{N}(\hat{\theta}^{k-1}, \Sigma_{\theta_{k-1}})$. In this case, the MAP estimate is given by

$$\hat{\theta}^k = \arg\min_\theta \|\tilde{\mathbf{y}}_k - m(\theta)\|_{\Sigma_e^{-1}}^2 + \|\theta - \hat{\theta}^{k-1}\|_{\Sigma_{\theta_{k-1}}^{-1}}^2. \quad (11)$$

Furthermore, $p_{\theta|k}(\theta|k)$ is often approximated as a gaussian variable with mean $\hat{\theta}^k$ and variance

$$\Sigma_{\theta_k} = \left(\Sigma_{\theta_{k-1}}^{-1} + \overbrace{\frac{\partial m(\theta)}{\partial\theta}^\top \Sigma_e^{-1}\frac{\partial m(\theta)}{\partial\theta}\bigg|_{\theta=\theta_{k-1}}}^{I_{\theta_k}}\right)^{-1} \quad (12)$$

were I_{θ_k} is an approximation of the *information matrix* related to the experiment k.

In our case, the observations $\tilde{\mathbf{y}}_k$ are the vectors $\tilde{\mathbf{T}}_k, \tilde{\mathbf{C}}_k$. We have already noted that in our model only the dynamics $F_{CT}(T, \theta)$ from T to C depends on the unknown parameter θ. Furthermore, the measurement $\tilde{\mathbf{C}}_k$ is the sum of $\mathbf{C}_k$ and the white measurement noise $\mathbf{e}_{\mathbf{C}k}$, whose statistical properties are known. Therefore, we could apply Equations (11) and (12) in this framework if we knew the temperature $\mathbf{T}_k$ using $m(\theta) = F_{CT}(\mathbf{T}_k, \theta)$ and $\tilde{\mathbf{y}}_k = \tilde{\mathbf{C}}_k$. However, the corrupted signal $\tilde{\mathbf{T}}_k$ is available in our case leading to an *Errors-In-Variables* identification problem [10]. In this work we ignore the effect or the measurement error $\mathbf{e}_{\mathbf{T}k}$ on the temperature, since its variance is relatively small. In order to further reduce the influence of the measurement error, the high frequency components of this signal are removed from $\tilde{\mathbf{T}}_k$ through low-pass filtering and a sequence $\tilde{\mathbf{T}}_{k,f}$ is obtained. The actual temperature $\mathbf{T}_k$ is indeed not likely to contain high frequency components.

This suggests the use of Equations (11) and (12) for parameters update with $m(\theta) = F_{CT}(\tilde{\mathbf{T}}_{k,f}, \theta)$ and $\Sigma_e = \sigma_C^2 I^N$ where $\tilde{\mathbf{T}}_{k,f}$ is a filtered version of the vector $\tilde{\mathbf{T}}_k$.

Based on the previous considerations, this algorithm has been devised. At each batch k, these steps are executed:

1) The temperature reference $\mathbf{T}_k^r$ is set as the input to the temperature controller and the noisy measurements $\tilde{\mathbf{C}}_k, \tilde{\mathbf{T}}_k$ are collected.
2) The vector $\tilde{\mathbf{T}}_{k,f}$ is obtained by filtering $\tilde{\mathbf{T}}$ through the same filter $B(q)$ defined in the ILC algorithm.
3) A corrected model of the dynamics from T^r to S is found as $\mathbf{S} = \hat{S}_k(\mathbf{T}^r) = F_{ST^r}(\mathbf{T}^r, \hat\theta^k)$. The updated parameter $\hat\theta^k$ is computed as

$$\hat\theta^k = \arg\min_{\theta\in\mathbb{R}^4}\big(\|\tilde{\mathbf{C}}_k - F_{CT}(\tilde{\mathbf{T}}_{k,f}, \hat\theta^k)\|_{\Sigma_e^{-1}}^2 + \\ + \|\theta - \hat\theta^{k-1}\|_{\Sigma_{\theta_{k-1}}^{-1}}^2\big) \quad (13)$$

where $\Sigma_e = \sigma_C^2 I^N$ and $\Sigma_{\theta_k}^{-1} = \Sigma_{\theta_{k-1}}^{-1} + I_{\theta_k}$. I_{θ_k} is an approximation of the information matrix computed as

$$I_{\theta_k} = \frac{\partial F_{CT}(\tilde{\mathbf{T}}_{k,f}, \theta)}{\partial\theta}^\top \Sigma_e^{-1}\frac{\partial F_{CT}(\tilde{\mathbf{T}}_{k,f}, \theta)}{\partial\theta}\bigg|_{\theta=\theta_{k-1}}. \quad (14)$$

and $\Sigma_{\theta_0}^{-1}$ is set to 0 for simplicity.
4) The corrected model is used to compute the temperature profile for the next iteration

$$\mathbf{T}_{k+1}^r = \arg\min_{\mathbf{T}^r\in\mathbb{R}^N}\|\overline{\mathbf{S}}_{k+1} - \hat{S}_k(\mathbf{T}^r)\|^2. \quad (15)$$

The optimization problems (13) and (15) are solved numerically using the active-set method of the Matlab function `fmincon`. The same single shooting strategy as in the case of problem (9) is used for problem (15). For problem (13), derivatives of the objective function with respect to model parameters are computed analytically by integrating the sensitivity equations along with the model equations [8].

Note that the parameter estimation is performed during the controlled batch experiments. The effect of the disturbance δ_T on T provides excitation to the system along the trajectory that is followed in the controlled experiments. The input might not be informative enough to estimate all model parameters very accurately. However, the parts of the dynamics having the strongest influence on the control objective are naturally emphatized in such datasets. As a consequence, performing the estimation based on those datasets tends to provide good models for the particular control application. Analogous situations have been analyzed in the field of *Identification for Control* for the estimation of linear transfer function. It was shown that if the final use of the model is the design of a closed-loop controller, performing the same estimation on closed-loop data is significantly advantageous. These observations leaded to the concept of *iterative design*, where successive steps of closed-loop identification and model-based controller design are performed [4].

IV. SIMULATION RESULTS

In this section we evaluate the performance of the B2B scheme described in the previous section on two different test cases, namely in presence of parametric model mismatch only (Case 1) and in presence of both parametric and
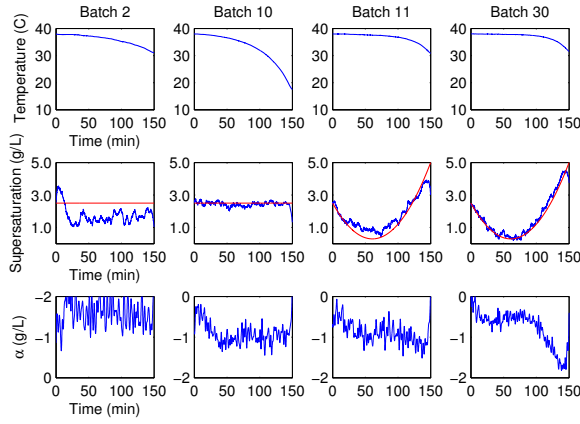
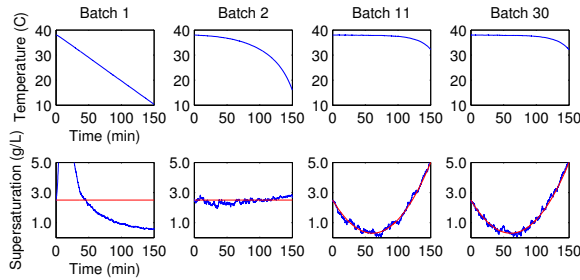Fig. 2.   Case 1 ILC: Batches 2,10,11,30

Fig. 4.   Case 2 ILC: Batches 2,10,11,30

Fig. 3.   Case 1 IIC: Batches 1,10,11,30

Fig. 5.   Case 2 IIC: Batches 2,10,11,30

structural mismatches (Case 2). The batch time is $t_f = 150$ min and the total number of batches is $N_b = 30$. The supersaturation set-point is the constant value $\overline{S}_k = 2.5$ g/L for the first 10 batches and is changed from batch 11 to 30 to a parabola passing through the points $(t, S) = \{(0, 2.5), (100, 1.2), (150, 5)\}$ (min, g/L).

The temperature reference at the first iteration $\mathbf{T}_1^r$ is set to a linear cooling from 38 to 10 ^oC in the time of the batch. The corresponding supersaturation is far away from the set-point throughout the batch (Figure 3, Batch 1).

For the ILC algorithm, the tuning of the scalar a_k is done as follows: $a_k = \overline{a}_k + c$ with $c = 0.1, \overline{a}_0 = 2$ and $\overline{a}_{k+1} = 2$ if a set-point change occurs at iteration $k+1$, $\overline{a}_{k+1} = 0.7\overline{a}_k$ otherwise.

Case 1

In this simulation the performance of the algorithms in presence of parametric model mismatch is evaluated. The true parameters used in the simulation model are $\theta_0 = [12\ 1.4\ 4\ 1]^\top$, while the nominal value $\hat{\theta} = [10.57\ 1.7\ 5\ 1.1]^\top$ is used by the B2B algorithms.

Results for the simulation are shown in Figure 2 for ILC and Figure 3 for IIC. In Batch 2 the tracking error is already small for IIC, while it is still appreciable for ILC. After some iterations also ILC approaches the set-point more closely (Batch 10). The performance is sensibly degraded in Batch 11 for ILC due to the set-point change, and is recovered again in the following iterations (Batch 30). Note also that the vector $\boldsymbol{\alpha}$ is smoothened during the iterations owing to
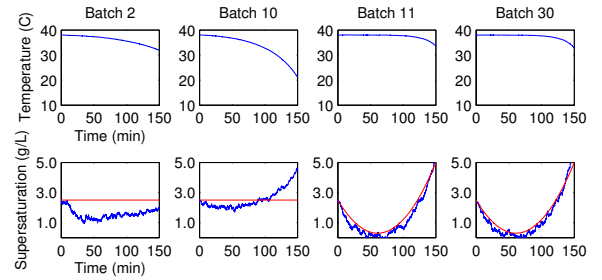
the Kalman filtering. In the IIC case, the set-point change does not lead to any performance loss in Batch 11.

Case 2

On top of the parametric mismatch already considered in Case 1, a structural model mismatch is introduced. A temperature-dependent crystal growth mechanism is assumed in the true system. The state equations 2,3,4 of $\mathcal{F}(x)$ in (2) are multiplied by a term $A_0 \exp(-E_a/R(T + 273.15))$, with $A_0 = 1.3 \times 10^7$, $E_a = 4.2 \times 10^4$ and $R = 8.3144$. This kind of temperature dependence is known as Arrhenius-type in literature [5].

The results of this simulation are reported in Figure 4 for ILC and Figure 5 for IIC. The IIC algorithm leads to a poor tracking performance in the first 10 batches (Batch 1, 10). In Batch 11 a better result is achieved. The cause of the better result is that the tracking of the new set-point requires a lower temperature variation compared to the previous case. Therefore, the mismatch of the nominal model that does not incorporate the temperature-varying growth behavior is less detrimental in these conditions. Note that this better result is not caused by a learning mechanism. Indeed, no further improvement is obtained in the following batches (Batch 30). On the contrary the ILC algorithm is still capable to approach the set-point, even though more iteration are required (Batches 10, 30).

Note that the correction vector in ILC is again much smoothened in Batch 30. However, a different correction is performed. Compared to Case 1, $\boldsymbol{\alpha}$ is more negative
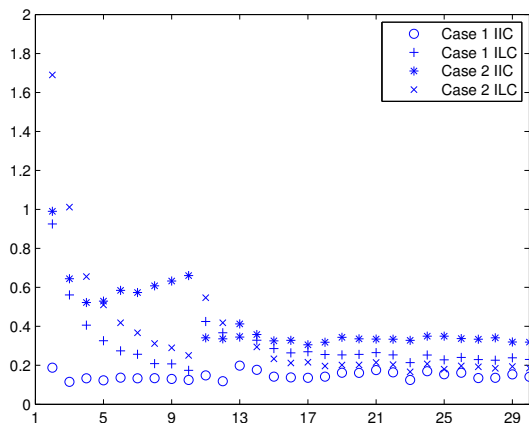
Fig. 6.   RMSE vs iteration number

at the beginning of the process and increases towards the end. This is reasonable from a physical point of view. At the beginning of the process, at high temperature, the temperature-dependent simulation model predicts a higher growth rate of crystals, thus also a higher supersaturation consumption. The nominal model, which does not incorporate this mechanism, has to be corrected to predict a lower supersaturation compared to the previous case. Towards the end of the process, at low temperature, the converse situation occurrs and the nominal model has to be corrected in order to predict higher supersaturation.

Overall Results and Discussion

The *Root Mean Square Error* (RMSE) of the supersaturation tracking error is plotted against the iteration number for all cases in Figure 6. The different behavior of the two algorithms is evident in this plot. IIC provides the best performance when the right model structure is assumed. A good result is already obtained after the first iteration of the algorithm and it is not considerably influenced by the set-point change, because the algorithm learns the full model structure (Case 1 IIC). However, the behavior of the algorithm is hard to predict in the case of structural model mismatches. The IIC framework indeed relies strongly on the assumption of the model structure. In the situation considered where a temperature-dependent growth behavior is incorporated in the simulation model, IIC leads to a worse overall performance compared to ILC. Furthermore, almost no improvements are obtained after some batches (Case 2 IIC). The performance seems to improve in Batch 11, but this is actually the effect of the set-point change. On the contrary, ILC is much more robust to model structure mismatches. Even though these mismatches make the convergence somewhat slower in Case 2, a satisfactory result is obtained after some batches. However, after a set-point change more iterations are needed in order to come close to the optimum again since the additive correction is a valid approximation of the real dynamics only locally around a particular trajectory.

The complexity of the two solutions has to be considered both in terms of the computational effort required and in terms of the ease of design. The design of the parameter update in IIC is rather straightforward by adopting the Bayesian approach. However, such estimation requires the solution of a nonlinear least-squares problem. On the other hand, the parameter update of ILC only requires matrix operations in order to compute the additive correction update. However, the tuning of the algorithm is more delicate and may still require some trial-and-error. The stochastic framework we introduced allows to make this procedure more transparent and through the single parameter a_k it is possible to vary the adaptation speed of the algorithm.

V. CONCLUSION

We have presented a batch-to-batch (B2B) solution for supersaturation control in batch cooling crystallization. The B2B controller drives the reference of the PI temperature controller in the so-called B2B+PI configuration. Two B2B algorithms are evalued in this paper, namely an Iterative Learning Control (ILC) and an Iterative Identification Control (IIC). The properties of the controlled system with the two solutions have been discussed and analyzed in a simulation study.

The algorithms are shown to have complementary advantages and disadvantages. For this reason, it would be useful to design a supervisory algorithm which could switch from one strategy to the other based on the results in the previous batches. As a second point the tuning of ILC, which has already been simplified in this paper, should be made more rigorous in future. As a final point the IIC algorithms needs to be adapted for the case in which slow batch-to-batch parameter variations are possible.

REFERENCES

[1] I. Chin, S. Qin, K. Lee, and M. Cho, "A two-stage iterative learning control technique combined with real-time feedback for independent disturbance rejection," *Automatica*, vol. 40, no. 11, pp. 1913–1922, 2004.

[2] M. Forgione, A. Mesbah, X. Bombois, and P. Van den Hof, "Iterative learning control of supersaturation in batch cooling crystallization," Accepted for *American Control Conference*, Montreal, Canada.

[3] M. Fujiwara, Z. Nagy, J. Chew, and R. Braatz, "First-principles and direct design approaches for the control of pharmaceutical crystallization," *Journal of Process Control*, vol. 15, no. 5, pp. 493–504, 2005.

[4] M. Gevers, "Identification for control: From the early achievements to the revival of experiment design," *European Journal of Control*, vol. 11, pp. 1–18, 2005.

[5] W. Luyben, *Chemical reactor design and control*. Wiley-Interscience, 2007.

[6] A. Mesbah, Z. Nagy, A. Huesman, H. Kramer, and P. Van den Hof, "Inonlinear model-based control of a semi-industrial batch crystallizer using a population balance modeling framework," *IEEE Transactions on Control Systems Technology, In press*, vol. In press, 2011.

[7] Z. Nagy, J. Chew, M. Fujiwara, and R. Braatz, "Comparative performance of concentration and temperature controlled batch crystallizations," *Journal of Process Control*, vol. 18, no. 3-4, pp. 399–407, 2008.

[8] H. Rabitz, M. Kramer, and D. Dacol, "Sensitivity analysis in chemical kinetics," *Annual review of physical chemistry*, vol. 34, no. 1, pp. 419–461, 1983.

[9] A. Randolph and M. Larson, *Theory of particulate processes*. Academic Press San Diego, CA, 1971.

[10] T. Soderstrom, "Errors-in-variables methods in system identification," *Automatica*, vol. 43, no. 6, pp. 939–958, 2007.