Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

ELSEVIER

Journal of Process Control 15 (2005) 11–21

JOURNAL OF PROCESS CONTROL

www.elsevier.com/locate/jprocont

A batch-to-batch iterative optimal control strategy based on recurrent neural network models

Zhihua Xiong, Jie Zhang *

Centre for Process Analytics and Control Technology, School of Chemical Engineering and Advanced Materials, University of Newcastle, Newcastle upon Tyne NE1 7RU, UK

Received 6 June 2003; received in revised form 22 March 2004; accepted 16 April 2004

Abstract

A batch-to-batch model-based iterative optimal control strategy for batch processes is proposed. To address the difficulties in developing detailed mechanistic models, recurrent neural networks are used to model batch processes from process operational data. Due to model-plant mismatches and unmeasured disturbances, the calculated optimal control profile may not be optimal when applied to the actual process. To address this issue, model prediction errors from previous batch runs are used to improve neural network model predictions for the current batch. Since the main interest in batch process operation is on the end of batch product quality, a quadratic objective function is introduced to track the desired qualities at the end-point of a batch. Because model errors are gradually reduced from batch-to-batch, the control trajectory gradually approaches the optimal control policy. The proposed scheme is illustrated on a simulated methyl methacrylate polymerisation reactor.

Keywords: Batch-to-batch control; Iterative learning control; Optimisation; Recurrent neural networks; Batch processes; Polymerisation

1. Introduction

Batch processes are suitable for the manufacturing of high value added products such as pharmaceuticals and special polymers. There have been growing interests in batch process optimisation in recent years [1,2]. To achieve good optimal control performance, it is very important to obtain an accurate model capable of providing accurate long range predictions. Development of comprehensive mechanistic models for polymerisation reactors to predict the polymer quality in terms of process operating conditions has shown to be a very important methodology to the efficient control of product quality [3,4]. Although first principle models have the advantage of being valid over a wide range of process operations, building such models often requires physical insight into the batch processes and a large amount of time and resources. As mentioned by Terwiesch et al. [1], the effort for building a useful kinetic model that is still just valid for a limited range of operating conditions often exceeds one man year. The long time scale in detailed mechanistic model development may make these models infeasible for agile responsive manufacturing, where the products are typically short-lived and of small-volume. To address this issue, empirical models based on process operational data can be utilised. Empirical models can generally be developed very quickly without requiring detailed insight into the processes, though they are usually not as reliable as mechanistic models. Several types of data based empirical models have been applied to polymerisation reactors including multivariate statistic approach [5] and neural network approach [6]. Neural networks have been shown to be able to approximate any continuous non-linear functions [7] and have been proposed as a promising tool for identifying empirical models [8]. If properly trained and validated, these neural network models can be used to predict steady-state and dynamic process behaviour reasonably well, hence, leading to improved process optimisation and control performance [9,10]. As a non-linear regression tool, neural networks have been increasingly used in modelling and control of

[*] Corresponding author. Tel.: +44-191-2227240; fax: +44-191-2225292.
E-mail address: jie.zhang@newcastle.ac.uk (J. Zhang).

0959-1524/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2004.04.005

chemical processes, especially complex non-linear processes where process understanding is limited [9,11,12]. In this paper, recurrent neural networks (RNN) are used to build empirical models capable of providing good long range predictions from batch process operational data.

In practice, the effort of batch process optimisation is often hampered by unknown disturbances, such as reactive impurities and model-plant mismatches. "Optimal on the model" may not necessarily mean "optimal on the process" due to the presence of unknown disturbances and model-plant mismatches. Since batch processes are of repetitive nature, it would be possible to use information of previous batches to improve the operation of the current batch. This is referred to as "batch-to-batch" or "run-to-run" optimisation [13–16]. Recently, batch-to-batch optimisation of operating conditions for improving product quality and/or process efficiency has generated a challenging area of research. Batch-to-batch optimisation exploits the repetitive nature of batch processes to determine the optimal operating policy [13]. The general idea of batch-to-batch optimisation is to use results from previous batches to find iteratively the optimal operating conditions, while performing the smallest number of sub-optimal runs and preferably no unacceptable ones. The key element in batch-to-batch optimisation is the way that information is extracted from previous batch runs and used to optimise the operation of subsequent ones [2,14].

Various strategies have been proposed for batch-to-batch optimisation in the literature. Tendency models [17,18] have been developed by using simplified reaction mechanism models to update the estimates of the model parameters at the end of each batch and then re-optimise the trajectories for the next batch. Yabuki and MacGregor [19] apply midcourse correction policies to product quality control in semi-batch reactors. Srinivasan et al. [14,20] present a novel approach based on characterising nominal solutions using a simplified theoretical model, which are then adjusted based on information obtained from measurements on new batches. Statistical correlation models and multivariate analysis techniques such as principal component analysis and partial least squares (PLS) [16] have also been proposed. Many strategies have been employed to compensate for modelling error using information from previous runs [13,21,22]. Dong et al. [21] use PLS-NN models to obtain input profiles that would achieve a target conversion and molecular weight in minimum time by solving a sequential quadratic programming (SQP) problem. They use batch-to-batch adaptation to overcome the problem of model errors. For control of polymer molecular weight distributions (MWDs), Clarke-Pringle and MacGregor [13] proposed a method to correct the manipulated variable set-point trajectories from batch-to-batch. The method uses errors between the measured and desired MWDs at the end of a batch to update the manipulated variable trajectories for the next batch. Doyle et al. [22] use batch-to-batch optimisation to achieve the desired particle size distribution (PSD) target in an emulsion polymerisation reactor. A simplified theoretical model is performed as predictor, but the prediction is corrected using an updated PLS model that relates the manipulated variables to the error from the theoretical model prediction and the measured distribution.

When the whole reference sequences for product qualities, $\mathbf{Y}_\mathrm{d} = (y_\mathrm{d}(t)),\ t \in (0, t_\mathrm{f})$, is available, iterative learning control (ILC) using optimisation can be introduced to directly update input trajectory [23–25]. ILC is a control technique that has been used extensively in the control of mechanical systems and is especially suited for repetitive processes because it uses previous tracking error signals to adjust the manipulated variable trajectories and/or initial conditions for the upcoming batch run. Lee and co-workers in several related articles [24,25] propose a model predictive control for batch processes (BMPC) approach with quadratic criterion for temperature control in batch processes. Effective tracking control performance is achieved despite model errors and disturbances. This ILC type approach may not be straightforwardly applied to product quality control in batch processes. The main interest in batch process operation is on the final product quality and/or quantity, and it is usually more difficult to set the reference trajectories for product qualities practically and reasonably than for reactor temperatures. Although the reference sequences of product qualities during a whole batch may not be obtained, the desired values of product qualities $y_\mathrm{d}(t_\mathrm{f})$ at the end time of a batch are usually known. This makes it still possible to improve the product qualities from batch-to-batch.

In this study, the idea of ILC is adopted to improve product quality from batch-to-batch. The RNN model predictions are iteratively modified by using errors of the RNN model during previous batch runs. Updated control policy is calculated for each batch using the modified model predictions. By such a means, model errors based upon modified predictions are gradually reduced with respect to the number of batches and control policy gradually approaches the optimal policy.

The rest of this paper is structured as follows. Section 2 introduces RNN for the modelling of batch processes and a method to iteratively modify the model predictions using model prediction errors of the previous batch runs. Section 3 presents a model-based iterative optimal control strategy. Application of this control strategy to a simulated batch methyl methacrylate (MMA) polymerisation reactor is given in Section 4. Finally Section 5 draws some concluding remarks.

2. Batch processes modelling using RNN

In this study, we consider batch processes where the batch run length (t_f) is fixed and divided into N equal intervals (i.e. $N = t_f/h$, with h being the sampling time) and all batches run from the same initial condition. The control variable is parameterised as a piecewise constant trajectory, i.e. the value of the control variable in each interval is kept constant. The control problem is to manipulate the input policy subject to given constraints so that the product qualities at the batch end satisfy specific desired values.

Let us define the input and product quality sequences as

$$U_k = [u_k(0), u_k(1), \ldots, u_k(N-1)]^T \quad (1)$$

$$\mathbf{Y}_k = [y_k(1), y_k(2), \ldots, y_k(N)]^T \quad (2)$$

where k is the batch index, $y \in R^n$ are product quality variables of batch processes, $u \in R^m$ is the input (manipulated) variable for the product qualities, and the initial conditions (u_0, y_0) are given.

In this study, RNN models are used to model the non-linear relationship between U_k and $\mathbf{Y}_k$. Fig. 1 shows the structure of a globally recurrent neural network for modelling a second order non-linear system. The lagged network output $\hat{y}(t)$ is fed back to the network input nodes as indicated by the back-shift operator q^{-1} i.e. $\hat{y}(t-1) = \hat{y}(t)q^{-1}$. In this way, dynamics are introduced into the network. The network output depends not only on the network inputs but also on the previous network outputs. Thus the predictions from a RNN model are long range or multi-step-ahead predictions. Given the initial conditions (u_0, y_0) and the input sequence U_k, a RNN model can predict recursively the output $\hat{y}_k(t_f)$ at the end of a batch. The network can be trained in the "back-propagation through time" fashion [26,27] using the Levenberg–Marquart optimisation algorithm to minimise its long-range prediction errors. Therefore a RNN model can usually offer much better long-range predictions than a feed forward neural network [28,29]. In the recurrent neural networks used in this paper, the sigmoidal function is utilised as the neuron activation function in the hidden layer whilst the activation function in the output layer neurons is linear. The network weights were initialised as random numbers uniformly distributed over the range (−0.1, 0.1). Cross-validation was utilised as a procedure to determine the network structure, such as the number of hidden neurons. Different networks were trained on the training data and the network with the least sum of squared errors (SSE) on the validation data was chosen as the best network. The generalisation performance was then assessed on the unseen testing data set.

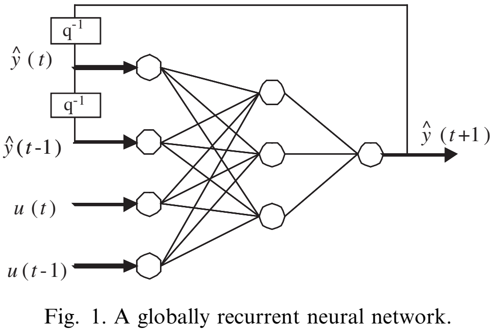

Fig. 1. A globally recurrent neural network.

As RNN models only approximate the batch process, prediction offsets can occur due to model-plant mismatches. After the completion of the current batch, the model prediction errors can be calculated by comparing the product quality measurement with the model prediction. The *measured* prediction error of RNN model for the kth batch, $\hat{e}_k(t)$, is defined as

$$\hat{e}_k(t) = y_k(t) - \hat{y}_k(t) \quad (3)$$

where $y_k(t)$ and $\hat{y}_k(t)$ are, respectively, the measured and predicted values of product quality variables at time t of the kth batch, and t is the discrete batch time.

Since batch processes are intended to be run repeatedly, it is reasonable to correct the model prediction $\hat{y}_{k+1}(t)$ at time t of the $(k+1)$th batch by adding the prediction errors $\hat{e}_k(t)$ at time t of the kth batch [13,15]. However, if significant noises are contained in the product quality measurement and/or significant disturbances exist and last for only one batch, an appropriate filter should be applied to the measurements to extract only the long term trend of the model prediction error. Doyle et al. [22] introduced a strategy to filter the model prediction residuals obtained from only one previous batch and the results show that this strategy can reduce batch-to-batch variability. Furthermore, as the dynamics of product quality variables in batch processes are usually non-linear, information of prediction errors in all previous runs should be utilised in deciding the input change for the next run. In this study, the average model errors of all previous runs are used to modify the RNN model predictions for the current batch. Because the batch process generally runs around the nominal trajectories, it is feasible to use the average errors to correct the RNN predictions. Using the averaged model prediction error to correct model predictions can reduce the effect of measurement noise and the effect of disturbances that only exist in one batch.

The average model error $\bar{\hat{e}}_k(t)$ of all previous runs is calculated as

$$\bar{\hat{e}}_k(t) = \frac{1}{k}\sum_{i=1}^{k}\hat{e}_i(t) = \frac{1}{k}\sum_{i=1}^{k}(y_i(t) - \hat{y}_i(t)) \quad (4)$$

By adding this average model error, the modified prediction $\tilde{y}_{k+1}(t)$ of RNN models is defined as

$$\tilde{y}_{k+1}(t) = \hat{y}_{k+1}(t) + \alpha\bar{\hat{e}}_k \quad (5)$$

where α is a bias correction parameter and $0 < \alpha \leqslant 1$. It should be noted that α cannot be set to 0. If $\alpha = 0$, without further modification to the model predictions, the same control policy will repeat in the subsequent batch run by solving the same optimisation problem. Large values of α could increase the accuracy of the modified RNN model prediction, but would also bring the significant changes of modified prediction if significant disturbances and/or uncertainties occur in the previous batch run, which will affect the convergence of iterative optimisation. So the trade-off between the convergence rate of iterative optimisation and the accuracy of the modified RNN model prediction should be considered.

3. Batch-to-batch model-based iterative optimal control

The formulation of the batch-to-batch model-based iterative optimisation control problem is as follows: use model-based optimisation to find an updating mechanism for the input sequence U_{k+1} of a new batch based on the information from previous runs so that the measured product qualities $y_{k+1}(t_f)$ at the batch end converges asymptotically towards the desired product qualities $y_d(t_f)$. However, the zero tracking error objective may not be satisfied for general batch processes. Since it is not uncommon for industrial batch processes to render a non-square problem for which zero tracking error for all the output variables is impossible, the zero tracking error requirement can be relaxed to minimum possible error in the least-squares sense [25]. Based on quadratic objective for batch-to-batch optimisation, the quality offsets can be eliminated as the run number is increased when there are enough degrees of input freedom and no random disturbances. When the input degrees of freedom are deficient, the offsets are minimised as dictated by the optimisation objective [25].

In the ILC approach, the input sequence can be obtained by a quadratic optimisation problem when the whole reference sequence $\mathbf{Y}_d$ is available. Both Amann et al. [23] and Lee et al. [25] have proposed to use the following quadratic objective in the ILC:

$$\min_{U_k} J = \|\mathbf{E}_k\|^2_{\mathbf{Q}} + \|U_k - U_{k-1}\|^2_{\mathbf{R}} \tag{6}$$

where $\mathbf{E}_k = \mathbf{Y}_d - \mathbf{Y}_k$ is the vector of tracking errors at the kth batch, $\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices. Because the input change is penalised instead of the input, the algorithm has an integral action with respect to the batch index k and achieves the minimum achievable error in the limit [25].

However, for product quality control of batch processes, only the desired product quality variables $y_d(t_f)$ at the end of a batch are of interest as mentioned above, then only errors at the end of a batch, $\tilde{e}^f_{k+1} = y_d(t_f) - \tilde{y}_{k+1}(t_f)$, are penalised in the objective function. Considering the constraints on the input trajectory, the batch-to-batch iterative optimisation problem for product quality control can be formulated as

$$\min_{U_k} J = \|\tilde{e}^f_k\|^2_{\mathbf{Q}} + \|\Delta U_k\|^2_{\mathbf{R}} \tag{7}$$

$$\text{s.t.}\quad \tilde{e}^f_k = y_d(t_f) - \tilde{y}_k(t_f) \tag{8}$$

$$\Delta U_k = U_k - U_{k-1} \tag{9}$$

$$\hat{y}_k(t) = \text{RNN}[\hat{y}_k(t-1), \ldots, u_k(t-1), \ldots] \tag{10}$$

$$\tilde{y}_k(t) = \hat{y}_k(t) + \alpha\bar{\hat{e}}_{k-1}(t) \tag{11}$$

$$u^{\text{low}} \leqslant U_k \leqslant u^{\text{hi}} \tag{12}$$

$$y^{\text{low}} \leqslant y_k(t_f) \leqslant y^{\text{hi}} \tag{13}$$

given $y_i(t), \forall t \in (0, t_f), i \leqslant k$.

In the above equations, $\tilde{e}^f_k$ is the difference between desired product qualities and modified predictions of the RNN model at the end of the kth batch, RNN[·] represents the RNN model, u^{low} and u^{hi} are the lower and upper bounds of the input trajectory, y^{low} and y^{hi} are the lower and upper bounds of the final product qualities, $\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices and they are selected of the following form in this study: $\mathbf{Q} = \lambda_q \cdot \mathbf{I}_n$ and $\mathbf{R} = \lambda_r \cdot \mathbf{I}_N$.

According to the objection function Eq. (7), the performance of the batch-to-batch iterative optimisation is affected by the weights λ_r and λ_q. A larger weight λ_r on the input change will lead to more conservative adjustments and slower convergence. The weight λ_q on the quality error term should be appropriately selected in relation to the weight λ_r so that the performance due to input changes will not be degraded while the product quality control is enforced. There are also other variants of the objective function. For example, the weight matrix $\mathbf{R}$ may be designed to be increasing with the batch number in proportion to the improved confidence of the product quality prediction.

The above optimisation approach considers constraints of the inputs and final product qualities in the form of linear inequalities. These constraints are very useful to define the feasible region for the optimisation problem when solving the optimisation problem using SQP. However, the lower and upper limits on the temperature may become active along the progression of solving quadratic programming problem, which will prevent the convergence to zero tracking error. The *path constraints* involving state variables in the form of nonlinear inequalities, such as $g[u_{k+1}(t), y_{k+1}(t)] \leqslant 0$, can also be considered. However, the computation cost of solving the optimisation problem could increase significantly when using the SQP approach.

The *modified* prediction error $\varepsilon_{k+1}(t)$ between measured product qualities and modified predictions is defined as

$$\varepsilon_{k+1}(t) = y_{k+1}(t) - \tilde{y}_{k+1}(t) \tag{14}$$

Considering Eq. (5), Eq. (14) can be rewritten as

$$\varepsilon_{k+1}(t) = \hat{e}_{k+1}(t) - \alpha\bar{\hat{e}}_k(t) \tag{15}$$

Eq. (4) can be reformulated as

$$\bar{\hat{e}}_k(t) = \frac{k-1}{k}\bar{\hat{e}}_{k-1}(t) + \frac{1}{k}\hat{e}_k(t) \tag{16}$$

As k approaches infinity, the above equation indicates that

$$\lim_{k\to\infty}\bar{\hat{e}}_k(t) = \lim_{k\to\infty}\frac{k-1}{k}\bar{\hat{e}}_{k-1}(t) + \lim_{k\to\infty}\frac{1}{k}\hat{e}_k(t) = \bar{\hat{e}}_\infty(t) \tag{17}$$

The neural network model predictions at the end of the $(k+1)$th batch can be approximated by the first order Taylor series approximation as

$$\begin{aligned}\hat{y}_{k+1}(t_\mathrm{f}) &= f_\mathrm{NN}(U_{k+1})\\ &= f_\mathrm{NN}(U_k) + \left[\left.\frac{\partial f_\mathrm{NN}}{\partial U}\right|_{U_k}\right]^\mathrm{T}\Delta U_{k+1}\end{aligned} \tag{18}$$

where f_NN is the non-linear function represented by the recurrent neural network, U is a vector of control actions in a batch, and $\Delta U_{k+1} = U_{k+1} - U_k$.

It is reasonable to assume that the iterative optimisation produces no worse solutions since the trivial solution $\Delta U_{k+1} = 0$ can ensue this. The following holds under this assumption

$$0 \leqslant \|\tilde{y}_{k+1}(t_\mathrm{f}) - y_\mathrm{d}\|_Q^2 \leqslant \|\tilde{y}_k(t_\mathrm{f}) - y_\mathrm{d}\|_Q^2 \tag{19}$$

It therefore follows that

$$\lim_{k\to\infty}\tilde{y}_{k+1}(t_\mathrm{f}) = \lim_{k\to\infty}\tilde{y}_k(t_\mathrm{f}) \tag{20}$$

It follows from Eqs. (5) and (18) that

$$\begin{aligned}\lim_{k\to\infty}\tilde{y}_{k+1}(t_\mathrm{f}) &= \lim_{k\to\infty}[\hat{y}_{k+1}(t_\mathrm{f}) + \alpha\bar{\hat{e}}_k(t)]\\ &= \lim_{k\to\infty}[f_\mathrm{NN}(U_K)\\ &\quad + \left(\left.\frac{\partial f_\mathrm{NN}}{\partial U}\right|_{U_k}\right)^\mathrm{T}\Delta U_{k+1} + \alpha\bar{\hat{e}}_k(t)]\\ &= \lim_{k\to\infty}\tilde{y}_{k+1}(t_\mathrm{f})\\ &\quad + \lim_{k\to\infty}\left(\left.\frac{\partial f_\mathrm{NN}}{\partial U}\right|_{U_k}\right)^\mathrm{T}\Delta U_{k+1}\end{aligned} \tag{21}$$

It follows from the above equation that

$$\lim_{k\to\infty}\Delta U_{k+1} = 0 \tag{22}$$

Therefore the iterative optimal control scheme converges with respect to the batch number k.

As shown in Fig. 2, the procedure of batch-to-batch model-based iterative optimal control to update the recipe based on off-line product quality measurements is outlined as follows:

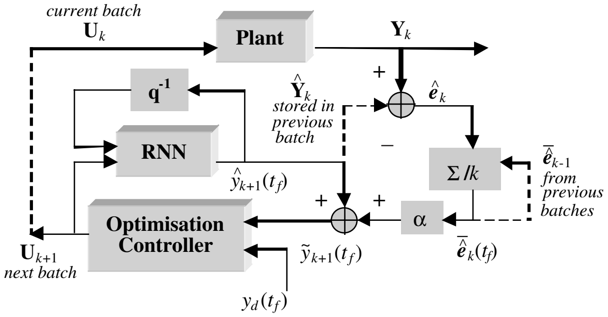

Fig. 2. Batch-to-batch iterative optimal control based upon model prediction modification.

(1) At the current kth batch, the input trajectory U_k is implemented into the batch process and the outputs $y_k(t)$ are measured either on-line or off-line after the completion of the current batch.
(2) Compare the measured $y_k(t)$ with the model prediction $\hat{y}_k(t)$ and calculate the model prediction error $\hat{e}_k(t)$ using Eq. (3) and store them. Using $\hat{e}_k(t)$ and those prediction errors stored in all previous runs to calculate the average model error $\bar{\hat{e}}_k(t)$ using Eq. (4).
(3) Modify the RNN model predictions for the next, the $(k+1)$th, batch and obtain $\tilde{y}_{k+1}(t)$ from Eq. (5). Based on the modified predictions $\tilde{y}_{k+1}(t_\mathrm{f})$, the quadratic optimisation problem specified by Eqs. (7)–(13) is solved and an updated open-loop input trajectory U_{k+1} for the $(k+1)$th batch is calculated.
(4) Increase the batch index k by 1 and go to Step 1.

4. Application to a batch polymerisation reactor

4.1. The batch polymerisation reactor and its control problem

The simulated batch polymerisation reactor studied here is based on a pilot scale polymerisation reactor installed at the Department of Chemical Engineering, Aristotle University of Thessaloniki, Greece. The reactor is shown in Fig. 3. The reaction is the free-radical solution polymerisation of MMA with a water solvent and benzoyl peroxide initiator. The reactor is provided with a stirrer for thorough agitation of the reacting mixture. Heating and cooling of the reacting mixture is achieved by circulating water at an appropriate temperature through the reactor jacket. The reactor temperature is controlled by a cascade control system consisting of a primary PID and two secondary PI controllers. The reactor temperature is fed back to the primary controller whose output is taken as the setpoint of the two secondary controllers. During the polymerisation, samples are taken, and the conversion is determined gravimetrically, while the number-average

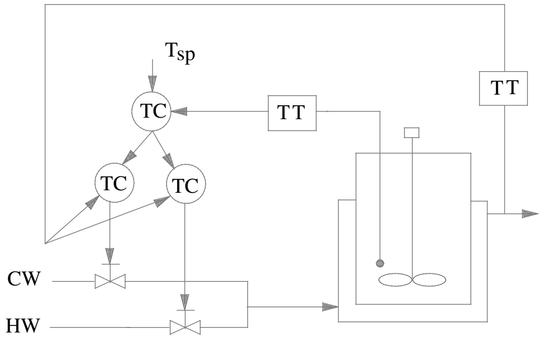

Fig. 3. A batch polymerisation reactor.

molecular weight (MN), weight average molecular weight (MW), and polydispersity are determined by gel permeation chromatography.

The general description of the chemical reactions during the free-radical solution polymerisation of MMA initiated by the benzoyl peroxide can be expressed as

Initiator decomposition: $I \xrightarrow{K_d} 2R_0$

Initiation: $R_0 + M \xrightarrow{K_i} R_1$

Propagation: $R_x + M \xrightarrow{K_p} R_{x+1}$

Transfer to monomer: $R_x + M \xrightarrow{K_m} P_x + R_1$

Transfer to solvent: $R_x + S \xrightarrow{K_s} P_x + R_1$

Termination by disproportionation:

$R_x + R_y \xrightarrow{K_{td}} P_x + P_y$

Termination by combination: $R_x + R_y \xrightarrow{K_{tc}} P_{x+y}$

In the polymerisation process, initiator I is decomposed into initiator radical R_0 The initiator radical R_0 reacts with monomer M and a radical R_1 of length 1 is generated. Monomer M is added onto the end of the radical R_x of length x, forming a new radical R_{x+1} of length $x+1$. The chain of radical R_x is transferred to monomer M and solvent S, forming dead polymers P_x and radicals R_1 of length 1. Termination by disproportionation generates polymers P_x and P_y, while termination by combination generates polymers P_{x+y}.

A detailed mathematical model covering reaction kinetics and heat mass balances has been developed for the bulk polymerisation of MMA [30]. Based on this model, a rigorous simulation program was developed and validated on the pilot reactor. In this study, the simulation programme is used as the real process to generate polymerisation data under different batch operating conditions and to test modelling and optimal control strategies. It is assumed in this study that the detailed mechanistic model is not available for optimisation control and data based empirical models have to be utilised. This is to mimic many practical situations where the development of detailed mechanistic models is not feasible for responsive manufacturing.

The optimal control problem for this batch polymerisation reactor is to find the optimal temperature profile through minimising a final-state performance index at a given final time. The performance index is given below

$$\mathrm{PI}(T_d) = \|\mathbf{y}_d(t_f) - \mathbf{y}(t_f)\|^2 \tag{23}$$

where $\mathbf{y} = [X, M_n, M_w]^T$, X is the monomer conversion, M_n is the dimensionless number-average molecular weight and $M_n = \mathrm{MN}/\mathrm{MN}_{ref}$, M_w is the dimensionless weight-average molecular weight and $M_w = \mathrm{MW}/\mathrm{MW}_{ref}$, T_d is the dimensionless temperature profile (as input variable) and $T_d = (T - T_{min})/(T_{max} - T_{min})$, $y_d(t_f)$ is the desired value for y at the end of batch which is set to $[1.0, 1.0, 1.0]^T$, and t_f is the final batch time which is set to 120 minutes in this study. The values of MN_{ref} and MW_{ref} used here are 3.0×10^5g/mol and 9.0×10^5 g/mol, respectively. The values of T_{min} and T_{max} used here are 310 and 360 K, respectively.

The polymer property constraint is on M_n:

$$0.95 \leqslant M_n(t_f) \leqslant 1.05 \tag{24}$$

Also the temperature profile is bounded by

$$0 \leqslant T_d \leqslant 1 \tag{25}$$

Due to model-plant mismatches, the ‘‘optimal control policy’’ can only guarantee the product quality constraint Eq. (19) being satisfied on the model and cannot guarantee it being satisfied on the real process. To address this issue, the constraint Eq. (24) is considered as a soft constraint, which is allowed to be violated to certain extent. The hard constraint on M_n is considered in this study as

$$0.9 \leqslant M_n(t_f) \leqslant 1.1 \tag{26}$$

The hard constraint should not be violated and the violation of a hard constraint represents off specification products being produced. The back-off between the soft and hard constraints reflects the extents of model-plant mismatches.

4.2. Modelling the batch polymerisation reactor using RNN

In this study, the above mechanistic model is assumed to be unavailable for process optimal control and RNN models are utilised to model the non-linear relationship between product quality variables (y) and temperature policy (T_d) from process operation data. From the performance index and constraints Eqs. (23)–(26), there are three important product quality variables, i.e. X, M_n and M_w, which should be considered in determining the optimisation control policy. To develop optimal control strategies, models for predicting these product quality variables are required. In this study, three recurrent neural networks are used to model X, M_n and M_w.

In this study, 31 different sets of temperature profiles have been randomly chosen within a reasonable range to generate 31 batches of simulation data. As the batch time is 120 minutes and the sampling time is 4 minutes, each set of data contains 30 data points. All data are scaled to dimensionless values as mentioned before. Normally distributed random noises with zero means were added to all the simulation data to represent the effects of measurement noises. The standard deviations of the noises are 0.012, 0.009 and 0.01 for the dimensionless X, M_n and M_w, respectively. The entire data set was divided into 3 parts, 25 sets of data were used for training, 5 sets of data for validation, and the remaining 1 set for testing.

It was tested that introducing monomer conversion X as one of the inputs of the other two neural networks was necessary. The best representations of the three RNN models selected through cross-validation can be summarised as the following forms

$$\widehat{X}(t) = \mathrm{RNN}_1[\widehat{X}(t-1), \widehat{X}(t-2), T_d(t-1), T_d(t-2), T_d(t-3)] \tag{27}$$

$$\widehat{M}_n(t) = \mathrm{RNN}_2[\widehat{M}_n(t-1), \widehat{M}_n(t-2), T_d(t-1), T_d(t-2), T_d(t-3), \widehat{X}(t-1), \widehat{X}(t-2)] \tag{28}$$

$$\widehat{M}_w(t) = \mathrm{RNN}_3[\widehat{M}_w(t-1), \widehat{M}_w(t-2), T_d(t-1), T_d(t-2), \widehat{X}(t-1), \widehat{X}(t-2)] \tag{29}$$

where $\widehat{X}$, $\widehat{M}_n$ and $\widehat{M}_w$ are model predictions of X, M_n and M_w, respectively. The numbers of hidden neurons for the above networks, Eq. (27) to Eq. (27), are 8, 12 and 10, respectively and they were determined through cross-validation. The SSE of the models for X, M_n and M_w on the validation data set are 0.0978, 0.1935 and 0.2033, respectively. Fig. 4 shows the long-range predictions of these RNN models on the unseen testing data set. It is clear that the RNN models have captured the dynamic trends of the product qualities in the data quite well.

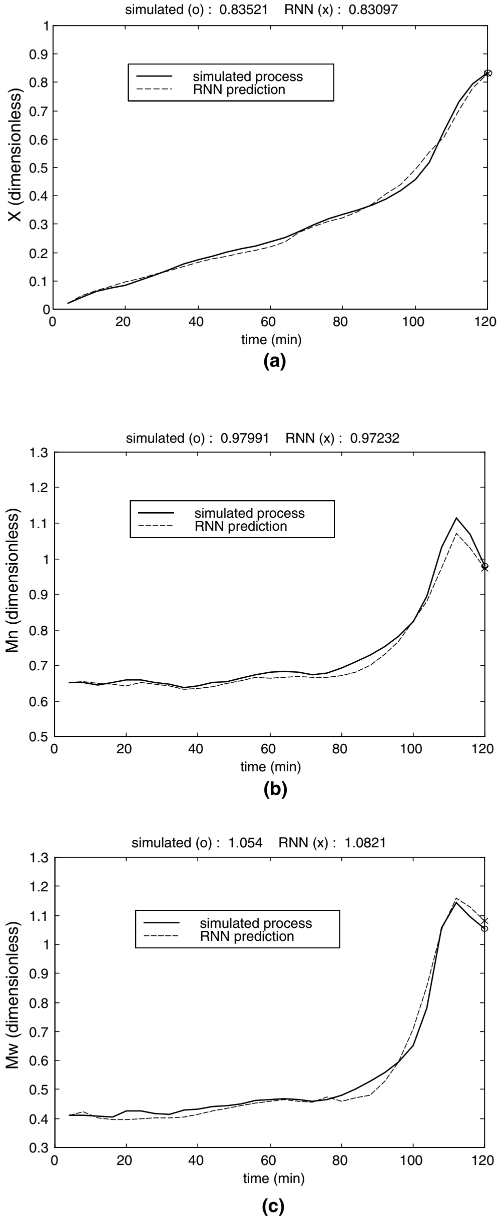

Fig. 4. Long-range predictions from RNN models for (a) X, (b) M_n, and (c) M_w.

4.3. Simulation results and discussions

To investigate the performance of the proposed control strategy for this batch polymerisation reactor, three cases were studied: Case 1—mechanistic model-based optimisation; Case 2—RNN model-based optimisation; and Case 3—RNN model-based batch-to-batch iterative optimisation. The optimisation problems in Case 1 and Case 2 are specified by Eqs. (23)–(26) based on the mechanistic model and the RNN models, respectively, whereas in Case 3 the optimisation problem is specified by Eqs. (7)–(13). Model errors in Case 2 were used as the initial condition for the model prediction modification in Case 3. All these non-linear optimisation problems were solved by the SQP method. The termination tolerance on the objective function was set to 10^{-6}. The batch length was divided into $N = 10$ equal intervals, and the temperature profile was parameterised as a piecewise constant trajectory correspondingly.

Different values of the parameters in the optimisation problem were considered. Large value of bias correction parameter α increases the accuracy of the modified RNN model prediction, but also brings the significant changes of modified prediction when disturbances occur during the procedure of iterative optimisation, which affects the convergence of iterative optimisation. After considering the trade-off between the convergence rate of iterative optimisation and the accuracy of the modified RNN model prediction, α was chosen as 0.25 in this study. The weighting factors in Eq. (7) were chosen as $\lambda_q = 3$ and $\lambda_r = 10$.

Batch processes always exhibit batch-to-batch variations due to unknown disturbances such as variations in raw material properties, reactive impurities, catalyst activities, and so on [1,20,31]. These kinds of disturbances always cause the variations of reaction parameters, for example, mass-transfer coefficients or kinetic parameters. In this study, unknown disturbances are simulated by reducing the initial initiator weight from its nominal value 2.5 g to 2.1 g in order to simulate the effect of reactive impurities [31]. It is assumed here that the unknown disturbances occurred from the 16th batch onward. Consequently, the reactive impurities act as a batch-wise persisting disturbance.

The product quality variables at the batch end and the performance index (PI) on the simulated process in all three cases without reactive impurities are shown in Table 1. It should be noticed that PI in Case 3 was obtained from Eq. (23) after the optimal temperature policy in Case 3 was applied to the simulated reactor. Due to RNN model-plant mismatches, the actual PI in Case 2, 0.0493, exceeds that in Case 1 by 45.8%. After 15 batches in Case 3, the performance index is decreased to 0.0382, only 13.0% higher (worse) than that in Case 1. It can also be seen from Table 1 that hard constraint on product quality is not satisfied in Case 2 due to the large model-plant mismatches. However, the hard constraint on product quality is satisfied in Case 3 through using iterative optimal control.

Fig. 5 compares the temperature profile of the 15th batch in Case 3 with the results in Case 1 and Case 2. Fig. 6 shows the temperature profiles in the 1st, 3rd, 11th and 15th batches of Case 3. Under the batch-to-batch iterative optimisation scheme, because model predictions were modified by the results of previous batches and model errors were gradually reduced, the control profile converges to the sub-optimal one. Fig. 7 shows that trajectories of the product qualities of X, M_n and M_w from Case 1 to Case 3 after the corresponding optimal temperature policies were applied to the simulated polymerisation reactor. It can be seen that after iterative optimisation control, the product qualities at the batch end in Case 3 are improved compared to those in Case 2.

Table 1
The product qualities and PI in all three cases without reactive impurities

	Case 1	Case 2	Case 3 (15th batch)
$X(t_f)$	0.8389	0.8321	0.8334
$M_n(t_f)$	0.9527	0.8902	0.9168
$M_w(t_f)$	0.9248	0.9046	0.9403
PI	0.0338	0.0493	0.0382

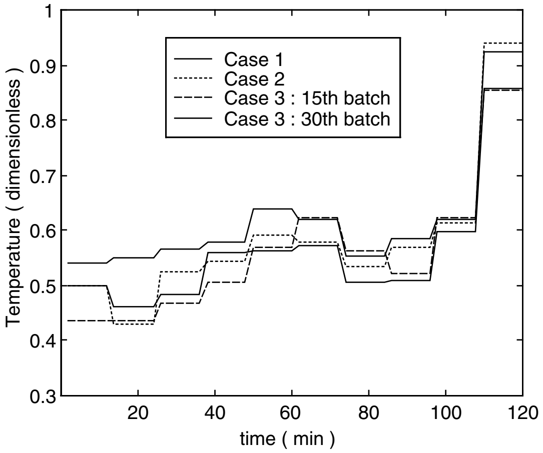

Fig. 5. Temperature profiles of Case 1, Case 2, and the 15th and 30th batches in Case 3.

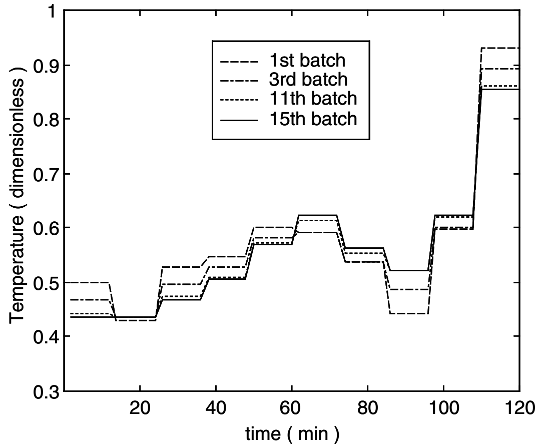

Fig. 6. Temperature profiles of the 1st, 3rd, 11th and 15th batches in Case 3.

Fig. 8 shows the convergence of the modified prediction errors $\varepsilon_k(t)$ of three product qualities with respect to the batch number during the first 15 batches. It is demonstrated in Fig. 8 that under the iterative optimisation scheme, almost all $\varepsilon_k(t)$ are gradually reduced from batch-to-batch. Fig. 9 shows the convergence of tracking errors $e_k^f = y_d(t_f) - y_k(t_f)$ of the three product quality variables during the first 15 batches in Case 3. It can be seen that after just 11 batches, tracking errors have converged with respect to the batch number. The tracking errors e_k^f does not converge to zero, because the RNN model-plant mismatches could still exist even though the RNN model predictions are corrected by measured product qualities. Furthermore, there are lower and upper limits on the temperature along the progression of solving SQP problem, which also prevents the convergence to zero tracking error.

When unmeasured disturbances (reactive impurities) occur in a batch, the optimal input trajectory calculated in the nominal case could be no longer appropriate. Table 2 shows the final product quality variables and PI on the simulated process under reactive impurities. With the presence of reactive impurities, (the initial initiator weight drops to 2.1 g from its nominal value 2.5 g), the PI in Case 1 and Case 2 become worse if their control policies are still employed. As shown in Table 2, the PI in Cases 1 and 2 increase to 0.0702 and 0.0917,

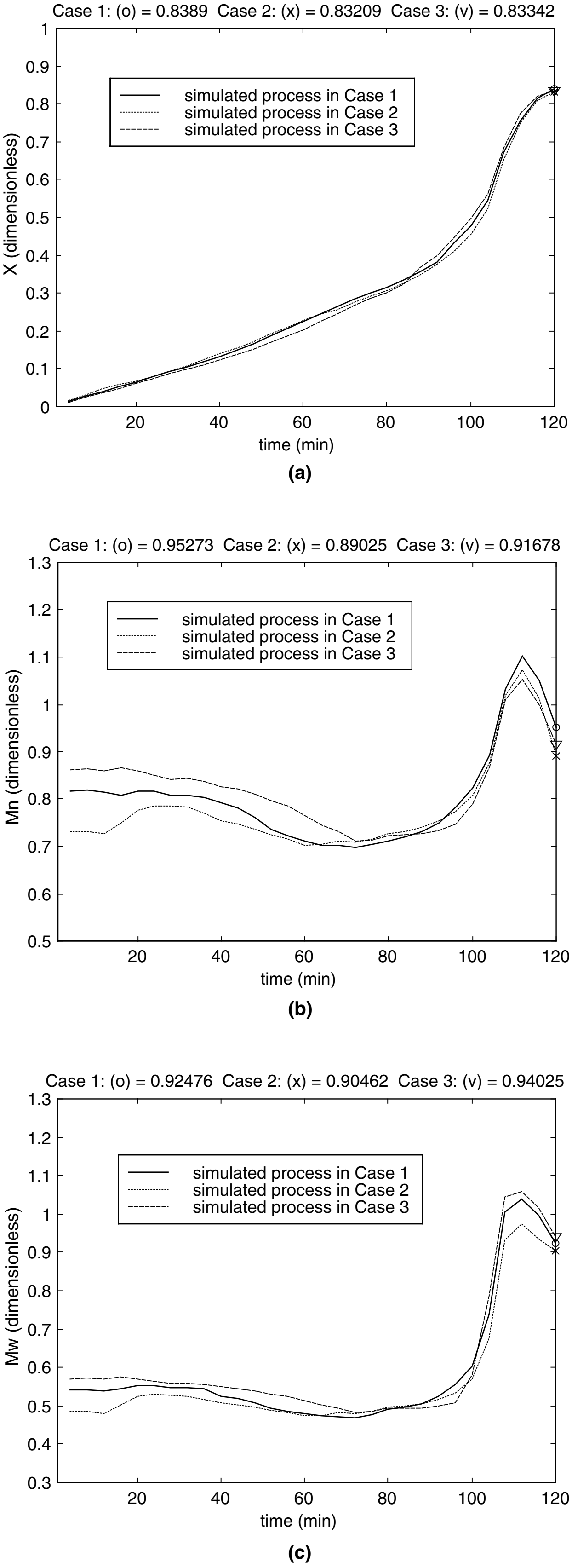

Fig. 7. Trajectories of three product quality variables on the simulated process: (a) X, (b) M_n, and (c) M_w.

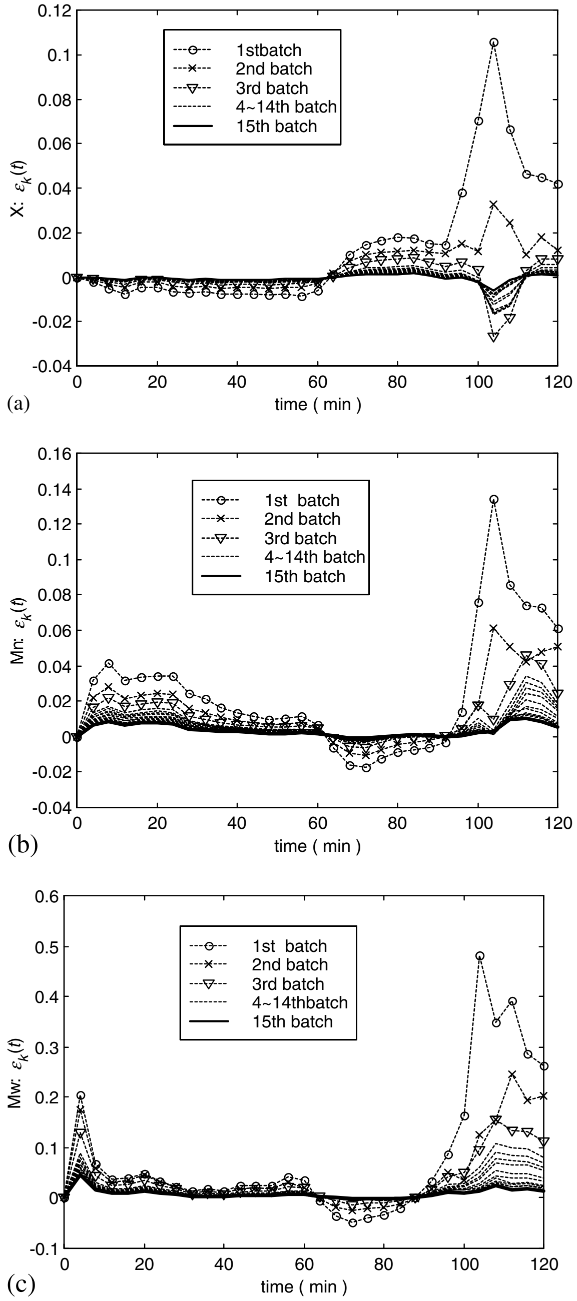

Fig. 8. Convergence of modified prediction errors $\boldsymbol{\varepsilon}_k(t)$ in Case 3: (a) X, (b) M_n, and (c) M_w.

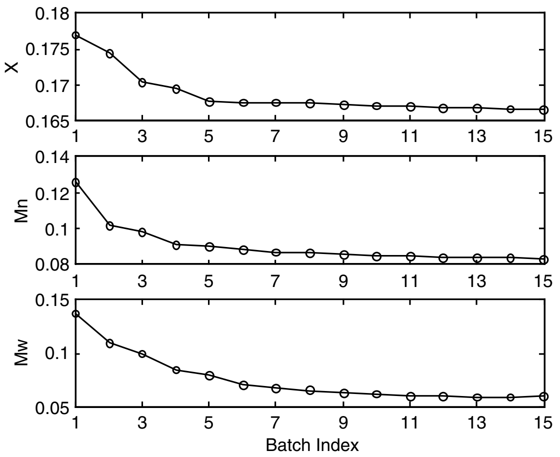

Fig. 9. Convergence of tracking errors e_k^f of three product quality variables in Case 3 without reactive impurities.

Table 2
The product qualities and PI in all three cases with reactive impurities

	Case 1	Case 2	Case 3 (30th batch)
$X(t_f)$	0.8335	0.8580	0.8481
$M_n(t_f)$	1.1354	1.1675	1.1010
$M_w(t_f)$	1.1553	1.2086	1.1161
PI	0.0702	0.0917	0.0467

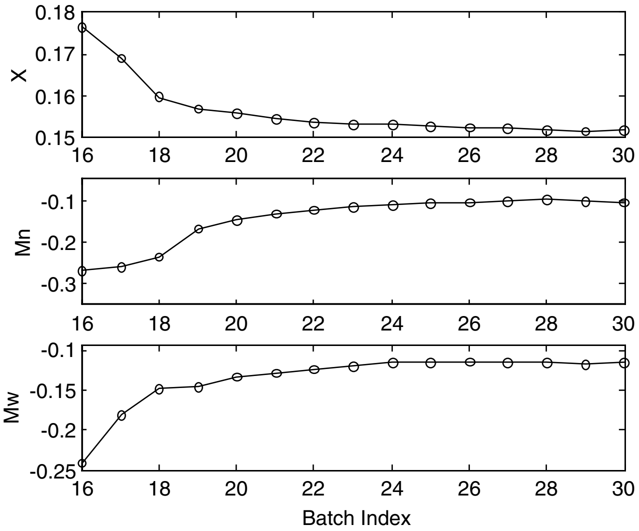

Fig. 10. Convergence of tracking errors e_k^f of three product quality variables in Case 3 under reactive impurities.

respectively. In Case 3, the temperature trajectory obtained in the 15th batch was not optimal any more for the 16th batch and prediction errors of RNN increased. The PI in the 16th batch is even degraded to 0.1598. As a consequence, tracking errors e_k^f of the 16th batch increased significantly, as shown in Fig. 10. Table 2 also shows that, without iterative optimal control, the optimal control action from the rigorous mechanistic model can lead to off specification product being produced when reactive impurities present. However, the detrimental effect of reactive impurities can be overcome by using the proposed iterative optimal control strategy.

Under the model-based batch-to-batch iterative optimisation scheme, because model prediction errors are gradually reduced, the tracking errors are reduced with respect to the batch index. Fig. 10 shows the convergence of tracking errors e_k^f during batches 16 to 30. It can be seen that after 10 batches all errors e_k^f have almost converged. The PI in the 30th batch of Case 3 is significantly reduced to 0.0467.

5. Conclusions

A model-based batch-to-batch iterative optimal control scheme for batch processes is proposed in order to overcome the detrimental effects of model-plant mismatches and unknown disturbances. A recurrent neural network model is built to represent the operation of a batch process from process operation data. The presence of model-plant mismatches and/or unknown disturbances is reflected by the neural network model prediction errors. To overcome the problem of model-plant mismatches and/or unknown disturbances, the model predictions are modified using model errors of previous batches. Based on the modified model predictions, optimal control policy is calculated by minimising a quadratic objective function concerning the product quality at the end of a batch. Using batch-to-batch iterative optimisation, it has been demonstrated that model errors are gradually reduced from batch-to-batch and the control trajectory gradually approaches the optimal control policy. The proposed scheme is illustrated on a simulated MMA polymerisation reactor. Simulation results demonstrate that the proposed approach can effectively overcome the problems of model-plant mismatches and unknown disturbances. Further studies on the effect of input parameterisation will be carried out in the future.

Acknowledgements

The authors gratefully acknowledge the support of the UK EPSRC (Grant GR/N13319—Non-linear Optimising Control in Agile Batch Manufacturing) and thank Prof. C. Kiparissides of the Aristotle University of Thessaloniki, Greece, for providing the polymerisation reactor model and the simulation programme. The authors thank the anonymous reviewers for their constructive comments which helped to improve the quality and presentation of this paper.

References

[1] P. Terwiesch, M. Argarwal, D.W.T. Rippin, Batch unit optimization with imperfect modelling: a survey, J. Process Control 4 (1994) 238–258.

[2] D. Bonvin, Optimal operation of batch reactors—a personal view, J. Process Control 8 (1998) 355–368.

[3] M.A. Dube, A. Penlidis, R.K. Mutha, W.R. Cluett, Mathematical modelling of emulsion copolymerization of acrylonitrile/butadiene, Ind. Eng. Chem. Res. 35 (1996) 4434–4448.

[4] A. Tsoukas, M. Tirrell, G. Stephanopoulos, Multiobjective dynamic optimisation of semibatch copolymerisation reactors, Chem. Eng. Sci. 37 (12) (1982) 1785–1795.

[5] J.F. MacGregor, C. Jaeckle, C. Kiparissides, M. Koutoudi, Process monitoring and diagnostics by multiblock PLS methods, AIChE J. 40 (1994) 826–838.

[6] J. Zhang, A.J. Morris, E.B. Martin, C. Kiparissides, Prediction of polymer quality in batch polymerisation reactors using robust neural networks, Chem. Eng. J. 69 (1998) 135–143.

[7] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Syst. 2 (1989) 303–314.

[8] J. Sjoberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson, A. Juditsky, Nonlinear black-box modeling in system identification: a unified overview, Automatica 31 (1995) 1691–1724.

[9] N. Bhat, T. McAvoy, Use of neural nets for dynamic modelling and control of chemical process systems, Comput. Chem. Eng. 14 (1990) 573–583.

[10] K.J. Hunt, D. Sbarbaro, R. Zbikowski, P.J. Gawthrop, Neural networks for control systems—a survey, Automatica 28 (1992) 1083–1112.

[11] I.M. Mujtaba, M.A. Hussain, Application of Neural Networks and Other Learning Technologies in Process Engineering, Imperial College Press, London, 2001.

[12] Y. Tian, J. Zhang, A.J. Morris, Modeling and optimal control of a batch polymerization reactor using a hybrid stacked recurrent neural network model, Ind. Eng. Chem. Res. 40 (2001) 4525–4535.

[13] T.L. Clarke-Pringle, J.F. MacGregor, Optimization of molecular-weight distribution using batch-to-batch adjustments, Ind. Eng. Chem. Res. 37 (1998) 3660–3669.

[14] B. Srinivasan, C.J. Primus, D. Bonvin, N.L. Ricker, Run-to-run optimization via control of generalized constraints, Control Eng. Pract. 9 (2001) 911–919.

[15] K. Lee, J.H. Lee, D.R. Yang, A.W. Mahoney, Integrated run-to-run and on-line model-based control of particle size distribution for a semi-batch precipitation reactor, Comput. Chem. Eng. 26 (2002) 1117–1131.

[16] J. Flores-Cerrillo, J.F. MacGregor, Within-batch and batch-to-batch inferential-adaptive control of semibatch reactors: a partial least squares approach, Ind. Eng. Chem. Res. 42 (2003) 3334–3345.

[17] C. Filippi-Bossy, J. Bordet, J. Villermaux, S. Marchal-Brassely, C. Georgakis, Batch reactor optimization by use of tendency models, Comput. Chem. Eng. 13 (1989) 35–47.

[18] A. Rastogi, J. Fotopoulos, C. Georgakis, G.H. Stenger, The identification of kinetic expressions and the evolutionary optimization of speciality chemical batch reactors using tendency models, Chem. Eng. Sci. 47 (1992) 2487–2492.

[19] Y. Yabuki, J.F. MacGregor, Product quality control in semibatch reactors using midcourse correction policies, Ind. Eng. Chem. Res. 36 (1997) 1268–1275.

[20] B. Srinivasan, D. Bonvin, E. Visser, S. Palanki, Dynamic optimization of batch processes. II. Role of measurements in handling uncertainty, Comput. Chem. Eng. 27 (2003) 27–44.

[21] D. Dong, T.J. McAvoy, E. Zafiriou, Batch-to-batch optimization using neural network models, Ind. Eng. Chem. Res. 35 (1996) 2269–2276.

[22] F.J. Doyle III, C.A. Harrison, T.J. Crowley, Hybrid model-based approach to batch-to-batch control of particle size distribution in emulsion polymerization, Comput. Chem. Eng. 27 (2003) 1153–1163.

[23] N. Amann, D.H. Owens, E. Rogers, Iterative learning control for discrete-time system with exponential rate of convergence, IEE Proc. D-Control Theory Appl. 143 (2) (1996) 217–224.

[24] K.S. Lee, I.S. Chin, H.J. Lee, J.H. Lee, Model predictive control technique combined with iterative learning control for batch processes, AIChE J. 45 (10) (1999) 2175–2187.

[25] J.H. Lee, K.S. Lee, W.C. Kim, Model-based iterative learning control with a quadratic criterion for time-varying linear systems, Automatica 36 (2000) 641–657.

[26] P.J. Werbos, Backpropagation through time: what it does and how to do it, Proc. IEEE 78 (1990) 1550–1560.

[27] H.T. Su, T.J. McAvoy, P. Werbos, Long term prediction of chemical processes using recurrent neural networks: a parallel training approach, Ind. Eng. Chem. Res. 31 (1992) 1338–1352.

[28] J. Zhang, A.J. Morris, E.B. Martin, Long-term prediction models based on mixed order locally recurrent neural networks, Comput. Chem. Eng. 22 (1998) 1051–1063.

[29] Y. Tian, J. Zhang, A.J. Morris, Optimal control of a batch emulsion copolymerisation reactor based on recurrent neural network models, Chem. Eng. Process. 41 (2002) 531–538.

[30] A. Penlidis, S.R. Ponnuswamy, C. Kiparissides, K.F. O'Driscoll, Polymer reaction engineering: modelling considerations for control studies, Chem. Eng. J. 50 (1992) 95–107.

[31] J. Zhang, A.J. Morris, E.B. Martin, C. Kiparissides, Estimation of impurity and fouling in batch polymerisation reactors through the application of neural networks, Comput. Chem. Eng. 23 (1999) 301–314.