IMPROVED OPERATION OF A BATCH POLYMERISATION REACTOR THROUGH BATCH-TO-BATCH ITERATIVE OPTIMISATION

Zhihua Xiong and Jie Zhang

Centre for Process Analytics and Control Technology
School of Chemical Engineering and Advanced Materials
University of Newcastle, Newcastle upon Tyne, NE1 7RU, U. K.
E-mail: zhihua.xiong@ncl.ac.uk, jie.zhang@ncl.ac.uk

Abstract: A batch-to-batch iterative product quality optimisation control strategy for a batch polymerisation reactor is proposed. Recurrent neural networks are used to model the dynamic behaviour of product quality variables. Model-plant mismatches and unknown disturbances are reflected in the model prediction errors. The repetitive nature of batch processes enables this information being discovered from previous batches and used to improve the current batch operation. Recurrent neural network predictions for the current batch are modified using prediction errors in previous batches. Because modified model errors are gradually reduced from batch to batch, the control trajectory gradually approaches to the optimal control policy and tracking errors also converge. The proposed scheme is illustrated on a simulated batch polymerisation reactor.

Keywords: iterative learning, optimisation, recurrent neural networks, batch processes, polymerisation.

1. INTRODUCTION

Batch-to-batch optimisation of operating conditions for improving product quality and/or process efficiency has generated a challenging area of research in batch processes. Batch-to-batch optimisation exploits the repetitive nature of batch processes to determine the optimal operating policy. The general idea of batch-to-batch optimisation is to use results from previous batches to find iteratively the optimal control policies for subsequent batches, while performing the smallest number of sub-optimal runs (Srinivasan *et al*., 2001). Various strategies have been proposed for batch-to-batch optimisation in the literatures. Some strategies have been employed to compensate for modelling error (Dong *et al.*, 1996; Crowley *et al.*, 2001). Recently, iterative learning control (ILC) using optimisation has been introduced to directly update input trajectory. Campbell *et al.* (2002) presented a brief survey of linear model based run-to-run control algorithms for batch processes. Lee and co-workers in several related articles (Lee *et al.*, 1999; Lee *et al.*, 2000) proposed the Q-ILC approach with quadratic criterion for temperature control of batch processes. It combines ILC with model predictive control. A linear time-varying model was built to represent reactor temperature in relation to feed and a pre-specified trajectory of the temperature was tracked. It has been shown that effective tracking control performance can be achieved despite model errors and disturbances. This approach demonstrates that based on a linear model and pre-specified trajectory, temperature control of batch processes can be carried out by an ILC type approach.

However, this ILC type approach is difficult to be used directly for product quality control of batch processes. It is usually more difficult to set the reference trajectories for product qualities $\mathbf{Y}_d = (\boldsymbol{y}_d(t))$, $t \in (0, t_f)$, practically and reasonably than for temperature. Even if such a reference trajectory can be set, it is usually difficult to measure how well the reference trajectory is tracked since many product quality variables are difficult to be measured on-line. Although the reference sequences of product qualities during a whole batch may not be obtained, the desired values of product qualities $\boldsymbol{y}_d(t_f)$ at the end

time of a batch are usually known. This makes it still possible to improve the product qualities from batch to batch. Furthermore, dynamics of product qualities cannot be represented accurately using a linear model since batch processes are operated in transient modes and their dynamics are intrinsically non-linear. The development of accurate mechanistic models of batch processes is usually costly and time-consuming. Alternatively, an empirical model, e.g. a recurrent neural network (RNN) model, can be built using process operation data to represent the non-linear dynamic characteristics of a batch process. In this study, RNN models are used to represent the non-linear relationship between the control trajectory and some product quality variables. The model predictions are iteratively modified by using model prediction errors in previous batches and optimisation is carried out based on the modified predictions.

The rest of this paper is structured as follows: Section 2 presents a batch-to-batch model-based iterative optimisation strategy. A simulated batch methyl methacrylate (MMA) polymerisation reactor is presented in Section 3. Section 4 gives simulation results of the proposed scheme on the MMA polymerisation reactor. Finally Section 5 draws some concluding remarks.

2. MODEL BASED BATCH-TO-BATCH ITERATIVE OPTIMISATION

2.1 Batch process modelling using recurrent neural networks

We consider a batch process where the run length (t_f) is fixed and divided into N equal intervals. Let us define the input and product quality sequences as

$$\mathbf{U}_k=[\boldsymbol{u}_k(0), \boldsymbol{u}_k(1),\ldots, \boldsymbol{u}_k(N\text{-}1)]^T \quad (1)$$

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

where k is the batch index, $\boldsymbol{y}\in R^n$ are product quality variables, $\boldsymbol{u}\in R^m$ are the input (manipulted) variables for controlling the product quality.

In this study, RNN models are used to model the non-linear relationship between $\mathbf{U}_k$ and $\mathbf{Y}_k$. Given the initial conditions ($\boldsymbol{y}_0$, $\boldsymbol{u}_0$) and the input sequence $\mathbf{U}_k$, RNN models can predict recursively the output $\hat{\boldsymbol{y}}(t_f)$ at the end of a batch. Thus the predictions from RNN models are long range or multi-step-ahead predictions. The networks are trained using the Levenberg-Marquart optimisation algorithm to minimise its long-range prediction errors. Therefore RNN models can usually offer much better long-range predictions than feed forward neural networks (Tian *et al.*, 2001).

The RNN model predictions can have errors due to model-plant mismatches and unknown disturbances. To reduce these errors, the RNN model predictions can be corrected by adding filtered model errors of previous batch runs. Crowley *et al.* (2001) introduced a strategy where the model predictions are corrected by adding the filtered model prediction residuals obtained from only the immediate previous run and showed that this can reduce batch-to-batch variability. As the dynamics of product quality in batch processes are usually very non-linear and measurement noise always exists, information of all previous runs should be used in updating model predictions for the next run. In this study, the average model errors of all previous runs are used to modify RNN model predictions. The RNN model error $\hat{\boldsymbol{e}}_k(t)$ is defined as

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

where $\boldsymbol{y}_k(t)$ and $\hat{\boldsymbol{y}}_k(t)$ are, respectively, the measured and predicted values of product quality at time t of the k^{th} batch. The average model error $\bar{\hat{\boldsymbol{e}}}_k(t)$ of all previous runs is defined as

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

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

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

where α is an adjustable filter parameter.

2.2 Batch-to-batch iterative optimisation control

Given that the whole reference sequence $\mathbf{Y}_d$ is available, both Amann *et al.* (1996) and Lee *et al.* (2000) have proposed a quadratic objective that penalises the input change 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 (Lee *et al.*, 2000). For product quality control of batch processes, only product qualities at the end of a batch, $\boldsymbol{y}_d(t_f)$, are available to set. Then only errors at the end of a batch, $\tilde{\mathbf{E}}^f_{k+1}=\boldsymbol{y}_d(t_f)-\tilde{\boldsymbol{y}}_{k+1}(t_f)$, 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

$$\min_{\mathbf{U}_{k+1}} J = \|\tilde{\mathbf{E}}^f_{k+1}\|^2_Q + \|\Delta\mathbf{U}_{k+1}\|^2_R \quad (6)$$

$$\text{s. t.} \quad \tilde{\mathbf{E}}^f_{k+1} = \boldsymbol{y}_d(t_f) - \tilde{\boldsymbol{y}}_{k+1}(t_f) \quad (7)$$

$$\Delta\mathbf{U}_{k+1}=\mathbf{U}_{k+1} - \mathbf{U}_k \quad (8)$$

$$\hat{\boldsymbol{y}}_{k+1}(t) =\boldsymbol{f}_{RNN}[\hat{\boldsymbol{y}}_{k+1}(t\text{-}1), \ldots, u_{k+1}(t\text{-}1), \ldots] \quad (9)$$

$$\tilde{\boldsymbol{y}}_{k+1}(t) = \hat{\boldsymbol{y}}_{k+1}(t) + \alpha\,\bar{\hat{\boldsymbol{e}}}_k(t) \quad (10)$$

$$u_{min} \le \mathbf{U}_{k+1} \le u_{max} \quad (11)$$

where $\tilde{\mathbf{E}}^f_{k+1}$ is the difference at the end of the $(k+1)^{\text{th}}$ batch between the desired product qualities and the modified RNN model predictions, $\boldsymbol{f}_{RNN}[\cdot]$ represents the RNN model, u_{min} and u_{max} are low and high bounds of the input trajectory, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are weighting matrices and they are selected of the following forms in this study: $\boldsymbol{Q}=\lambda_q\cdot\boldsymbol{I}_n$, and $\boldsymbol{R}=\lambda_r\cdot\boldsymbol{I}_N$.

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 $\boldsymbol{R}$ may be designed to be increaseing with batches reflecting the improved confidence of product quality prediction.

The modified prediction error $\varepsilon_{k+1}(t)$ is calculated as

$$\varepsilon_{k+1}(t) = \boldsymbol{y}_{k+1}(t) - \tilde{\boldsymbol{y}}_{k+1}(t) \qquad (12)$$

Considering Eq(5), Eq(12) can be rewritten as

$$\varepsilon_{k+1}(t) = \hat{\boldsymbol{e}}_{k+1}(t) - \alpha \bar{\hat{\boldsymbol{e}}}_k(t) \qquad (13)$$

Eq(4) can be reformed as

$$\bar{\hat{\boldsymbol{e}}}_k(t) = \frac{k-1}{k}\bar{\hat{\boldsymbol{e}}}_{k-1}(t) + \frac{1}{k}\hat{\boldsymbol{e}}_k(t) \qquad (14)$$

Then Eq(14) can be rewritten as

$$\varepsilon_{k+1}(t) = \frac{k-1}{k}\varepsilon_k(t) + [\hat{\boldsymbol{e}}_{k+1}(t) - \frac{k-1+\alpha}{k}\hat{\boldsymbol{e}}_k(t)] \qquad (15)$$

Beacasue $(k\text{-}1)/k < 1$, $\varepsilon_{k+1}(t)$ will converge with respect to the batch number k. Due to the reduced prediction errors of the RNN model, the control trajectory will gradually approach the optimal policy.

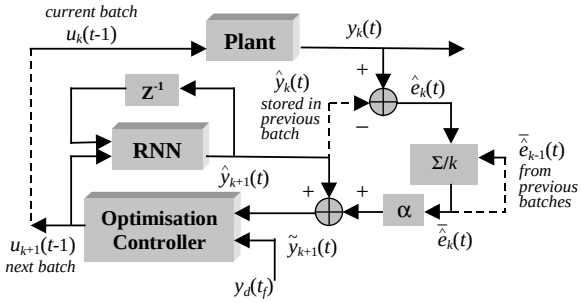

Fig. 1. Batch-to-batch iterative optimisation

As shown in Fig. 1, the batch-to-batch model-based iterative optimisation scheme is outlined as follows: At the current batch k, the input trajectory $\mathbf{U}_k$ is implemented and the outputs $\boldsymbol{y}_k(t)$ are obtained by on-line or off-line analysis of samples taken during the batch. The RNN model predictions for the next batch are modified by using prediction errors of all previous runs. Based on the modified predictions $\tilde{\boldsymbol{y}}_{k+1}(t)$, the quadratic optimisation problem specified by Eq(6) to Eq(11) is solved and a new control policy $\mathbf{U}_{k+1}$ for the next batch is calculated. This procedure is repeated from batch to batch. Because iterative optimisation is done in the interval between two adjacent batch runs, the outputs $\boldsymbol{y}_k(t)$, $t \in (1,N)$, could be obtained after the completion of each run by off-line analysis of samples taken during the batch.

3. A BATCH POLYMERISATION REACTOR

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 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. A detailed mathematical model covering reaction kinetics and heat mass balances has been developed for the bulk polymerisation of MMA. Based on this model, a rigorous simulation program was developed and validated on the pilot reactor. The simulation programme is used to test modelling and control strategies.

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

$$\boldsymbol{PI}(T) = \|\boldsymbol{y}_d(t_f) - \boldsymbol{y}(t_f)\|^2 \qquad (16)$$

where $\boldsymbol{y} = [X, Mn, Mw]^T$, X is the monomer conversion, Mn is the dimensionless number-average molecular weight (MN) and $Mn = \mathrm{MN}/\mathrm{MN}_{ref}$, MN_{ref} is set to 3.0×10^5 (g/mol), Mw is the dimensionless weight-average molecular weight (MW) and $Mw = \mathrm{MW}/\mathrm{MW}_{ref}$, MN_{ref} is set to 9.0×10^5(g/mol), T is the dimensionless temperature profile and $T = (T_r - T_{min})/(T_{rmax}-T_{min})$ with T_r being the unscaled temperature, T_{min} is 310K and T_{rmax} is 360K, $\boldsymbol{y}_d(t_f)$ is the desird value at the batch end 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 polymer property constraint is on Mn:

$$0.95 \le Mn \le 1.05 \qquad (17)$$

Also the temperature profile is bounded by

$$0 \le T \le 1 \qquad (18)$$

4. SIMULATION RESULTS AND DISCUSSIONS

RNN models are developed to model X, Mn and Mw. In this study, 31 different sets of temperature profiles have been randomly chosen within a reasonable range to generate 31 runs of simulation data. As the batch duration is 120 minutes and the sampling time is 4 minutes, each set of data contains 30 samples. All data are scaled to dimensionless values. 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, Mn and Mw respectively. The entire data set was divided into 3 parts, 25 batches of data were used for training, 5 batches of data for validation, and the remaining 1 batch for testing.

In the recurrent neural networks, hidden neurons use the sigmoidal activation function whilst output layer neurones use linear activation function. The network

weights were initialised as random numbers uniformly distributed over the range (-0.1, 0.1). The training algorithm is based on the Levenberg-Marquart algorithm. Network structures were determined through cross-validation. It was tested that introducing monomer conversion X as one of the inputs of the other two neural networks was necessary. 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. The best representations of the three RNN models selected through cross-validation are summarised as follows

$$\hat{X}(t) = f_1[\hat{X}(t\text{-}1), \hat{X}(t\text{-}2), T(t\text{-}1), T(t\text{-}2), T(t3)] \quad (19)$$

$$\hat{Mn}(t) = f_2[\hat{Mn}(t\text{-}1), \hat{Mn}(t\text{-}2), T(t\text{-}1), T(t\text{-}2), T(t\text{-}3), \hat{X}(t\text{-}1), \hat{X}(t\text{-}2)] \quad (20)$$

$$\hat{Mw}(t) = f_3[\hat{Mw}(t\text{-}1), \hat{Mw}(t\text{-}2), T(t\text{-}1), T(t\text{-}2), \hat{X}(t\text{-}1), \hat{X}(t\text{-}2)] \quad (21)$$

where $\hat{X}$, $\hat{Mn}$ and $\hat{Mw}$ are, respectively, the model predictions of X, Mn and Mw. The numbers of hidden neurons for the above three networks were determined through cross-validation as 8, 12 and 10 respectively. The SSE of the above models, Eq(19) to Eq(21), on the validation data set are 0.0978, 0.1935 and 0.2033 respectively. Fig. 2 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 quailities in the data quite well, though some prediction errors still exist.

To investigate the performance of the proposed control strategy, three cases were studied: Case 1 – optimisation based upon the mechanistic model; Case 2 – RNN model-based optimisation; and Case 3 – RNN model based batch-to-batch iterative optimisationp. In Case 1 and Case 2, the optimisation problem is specified by Eq(16) to Eq(18) based on the mechanistic model and the RNN models respectively, whereas in Case 3 it is specified by Eq(6) to Eq(11). All these non-linear optimisation problems were solved by the sequential quadratic programming method with the termination tolerance on objective function being set to 10^{-6}. Model errors in Case 2 are used as the initial condition for the model prediction modification in Case 3. Here the batch length was divided into N=10 equal intervals. The parameters in the optimisation problem in Case 3 were chosen as follows: α = 0.25, λ_q =3, and λ_r=10. Batch processes always exhibit batch-to-batch variations due to unknown disturbances such as reactive impurities and reactor fouling (Zhang *et al.*, 1999). In this study, the initial initiator weight was set to the nominal value 2.5g for the first 15 batches and then drops to 2.1g starting at the 16th batch to simulate the effect of reactive impurities. Consequently, the simulated reactive impurities act as a batchwise persisting disturbance.

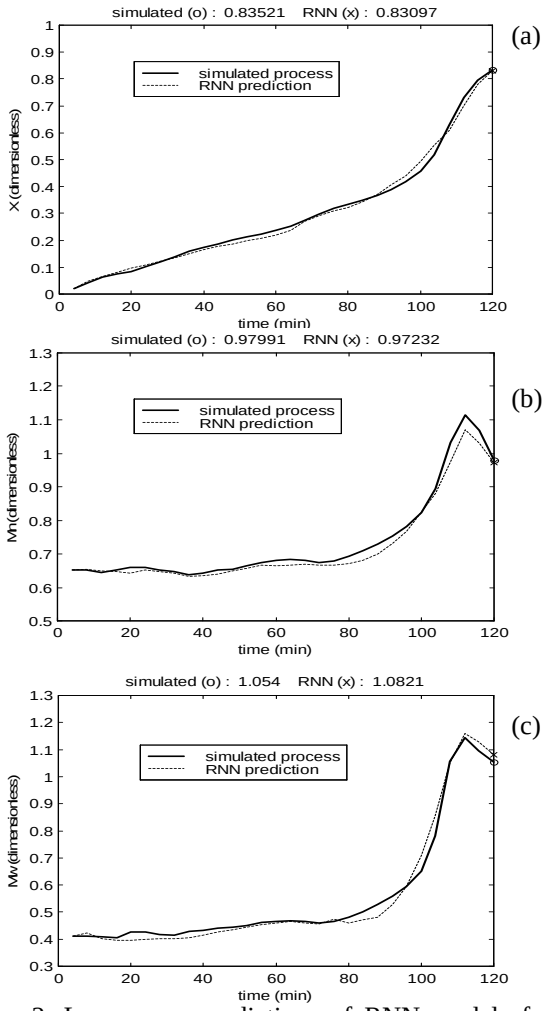

Fig. 2. Long-range predictions of RNN models for (a) X, (b) Mn, and (c) Mw

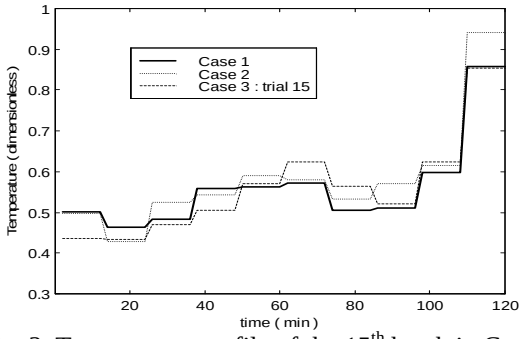

Fig. 3. Temperature profile of the 15th batch in Case 3 compared with those in Case 1 and Case 2

The results of the product qualities at the batch end and the performance indices (***PI***) in all three cases are shown in Table 1. It should be noticed that ***PI*** in Case 3 was obtained according to Eq(16) after the optimal temperature policy in Case 3 was applied to the simulated reactor. Due to RNN model-plant mismatches, the ***PI*** in Case 2, 0.0493, is 45.8% worse than that in Case 1. In Case 3, after 15 batches with iterative optimisation, the ***PI*** is decreased to 0.0382, only 13.0% higher (worse) than that in Case 1. Fig. 3 compares the temperature profile of the 15th batch in Case 3 with the results in Case 1 and Case 2.

Fig. 4 shows convergence of temperature profiles of the 1st, 3rd, 11th and 15th batches in Case 3. Under the proposed batch-to-batch iterative optimisation scheme, because model predictions were modified by the results of previous batches and model errors were gradually reduced, the reactor temperature profile also converge to the optimal one.

Table 1. The product qualities and ***PI*** in all three cases (without impurities)

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

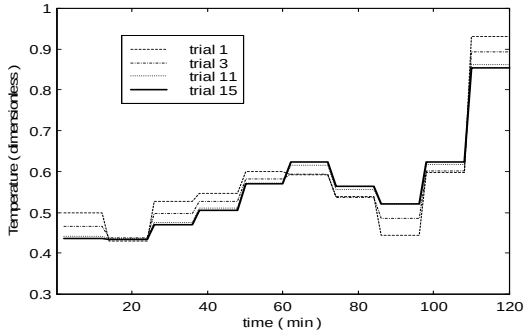

Fig. 4. Temperature profiles in Case 3

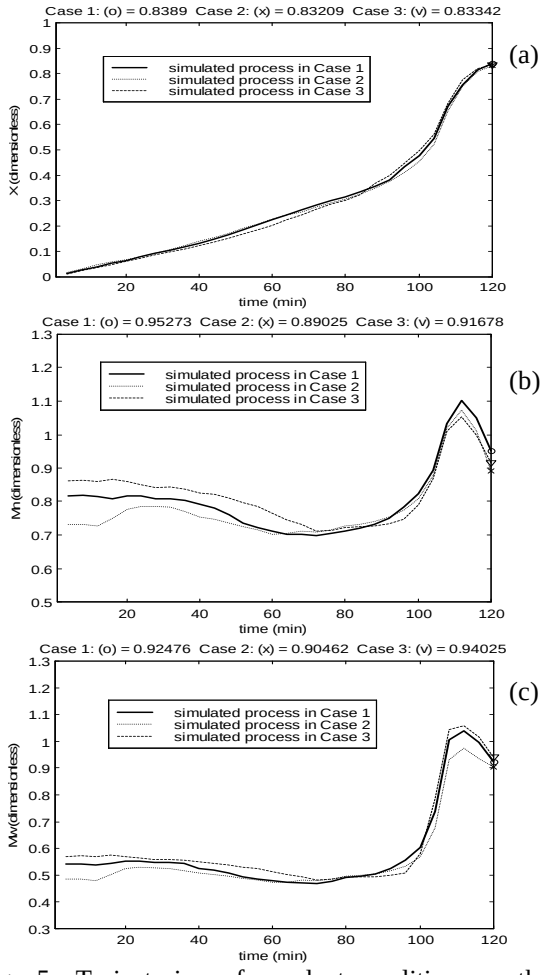

Fig. 5. Trajectories of product qualities on the simulated process: (a) *X*, (b) *Mn*, and (c) *Mw*

Fig. 5 shows the trajectories of *X*, *Mn* and *Mw* in three cases after the corresponding optimal temperature policies were applied to the simulated polymerisation reactor. It can be seen that with iterative optimisation control, the product qualities in Case 3 are improved compared to those in Case 2. Fig. 6 shows the convergence of tracking errors $\boldsymbol{e}_k^f = \boldsymbol{y}_d(t_f)-\boldsymbol{y}_k(t_f)$ of the product quality variables after the first 15 batches in Case 3. It can be seen that tracking errors have converged after about 11 batches.

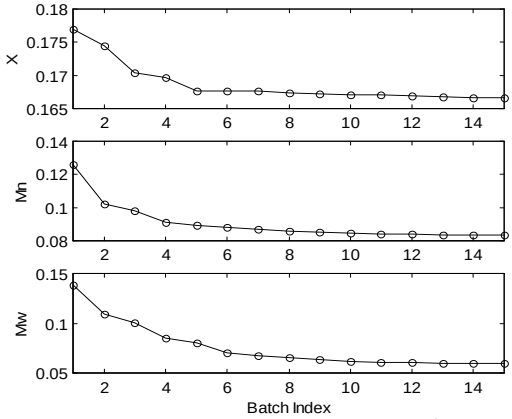

Fig. 6. Convergence of tracking errors $\boldsymbol{e}_k^f$ for the first 15 batches in Case 3

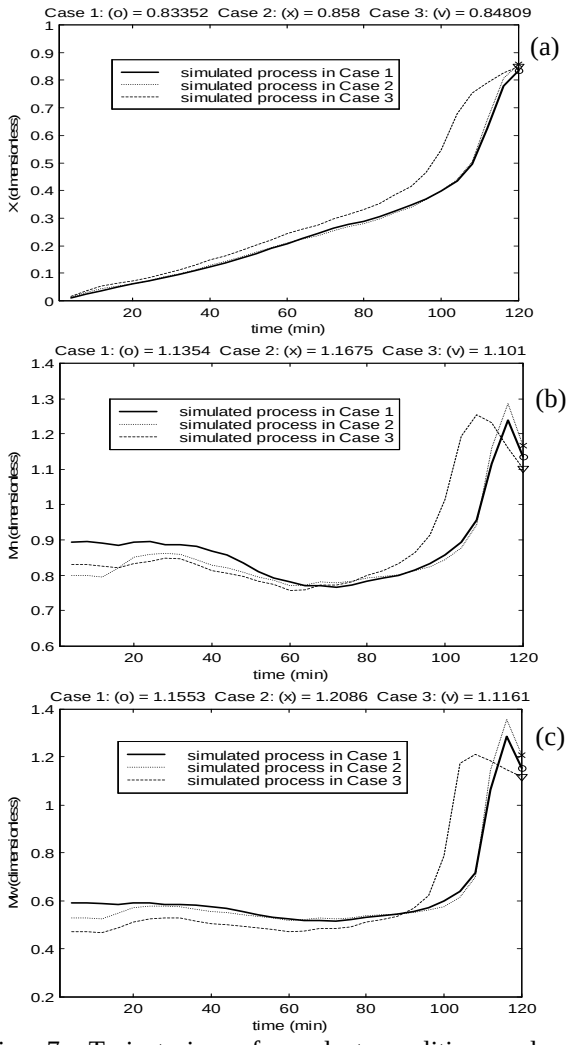

Fig. 7. Trajectories of product qualities under reactive impurities: (a) *X*, (b) *Mn*, and (c) *Mw*

If there are disturbances in a batch process, the optimal control profile calculated in the previous batch may not result in the expected product quality. When 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 the control policies calculated under the norminal condition are still employed. As shown in Table 2, the ***PI*** in Case 1 increases to 0.0702 and the ***PI*** in Case 2 increases to 0.0917. However, under the batch-to-batch iterative optimisation scheme, the ***PI*** in Case 3 can be brought down to 0.0467. Fig. 7 shows the trajectories of product quality variables under reactive impurities.

Due to the presence of reactive impurities, the temperature trajectory obtained in the 15th batch of Case 3 was no longer optimal and prediction errors of RNN increased. As shown in Fig. 8, the tracking errors e_k^f significantly increased in the 16th batch. This issue has been successfully addressed by the model-based iterative optimisation scheme. It can be seen that $\boldsymbol{e}_k^f$ has converged after about 10 batches. The results in Case 3 demonstrate that, although there are some model-plant mismatches and unknown disturbances, the performance index can be gradually improved under the iterative optimisation scheme.

Table 2. The product qualities and ***PI*** in all three cases (with unknown impurities)

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

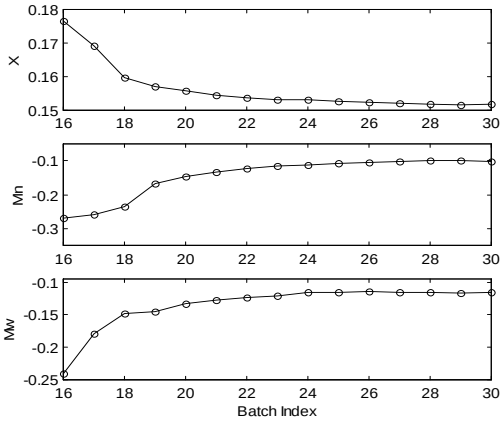

Fig. 8. Convergence of tracking errors $\boldsymbol{e}_k^f$ for the last 15 batches in Case 3

5. CONCLUSIONS

A model-based batch-to-batch iterative product quality optimisation control scheme for batch processes is proposed in this paper. Recurrent neural network models are built to represent the operation of a batch process and the model predictions are modified using prediction errors of previous batches. A quadratic objective function is introduced to the optimisation problem of product quality control. Using batch-to-batch iterative optimisation, it has been demonstrated that model errors are gradually reduced from batch to batch and the control policy converges to the optimal policy. The proposed scheme is illustrated on a simulated batch MMA polymerisation reactor.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the UK EPSRC (Grant GR/N13319) and thank Prof. C. Kiparissides of the Aristotle University of Thessaloniki, Greece, for providing the polymerisation reactor model and the simulation programme.

REFERENCES

Amann, N., D. H. Owens and E. Rogers (1996). Iterative learning control for discrete-time system with exponential rate of convergence. *IEE Proce. D -Control Theory Applications*, **143**(2), 217-224.

Campbell, W. J., S. K. Firth, A. J. Toprac and T. F. Edgar (2002). A comparison of run-to-run control algorithms. *Proce. ACC*, Anchorage, AK, May 8-10, 2150-2155.

Crowley, T. J., C. A. Harrison and F. J. Doyle III (2001). Batch-to-batch optimization of PSD in emulsion polymerization using a hybrid model. *Proce. ACC*, Arlington, VA, June 25-27, 981-986.

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

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

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

Srinivasan B., C. J. Primus, D. Bonvin and N. L. Ricker (2001). Run-to-run optimization via control of generalized constraints. *Control Engineering Practice*, **9**, 911-919.

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

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